

Large-Scale Display Interaction Techniques to Support Face-to-Face Collaboration

A thesis submitted
in partial fulfilment of the requirements
for the Degree of Master of Science
in Computer Science
in the University of Canterbury
by David Thompson

University of Canterbury

2006

ACKNOWLEDGEMENTS

To Pam and Brendon, for a life of learning and love.

To Jana and Becca, for understanding and for heart.

To Richard and Trond, for focus and inspiration.

To Vigo and Rob, for perspective and repair.

To Marilyn and Nathan, for hardware support.

To Mark and Tom, for the long view.

To Isobel, Tracey and Daniela, for making bureaucracy hurt less.

To Mr. Dick, Mr. Sterling, Mr. Gibson and Mr. Stephenson, for dreams.

I would not have reached here without you.

ABSTRACT

This research details the development of a large-scale, computer vision-based touch screen capable of supporting a large number of simultaneous hand interactions. The system features a novel lightweight multi-point tracking algorithm to improve real-time responsiveness. This system was trialled for six months in an exhibition installation at World Expo 2005 in Aichi, Japan, providing a robust, fault-tolerant interface.

A pilot study was then conducted to directly compare the system against other, more established input methods (a single-touch case, a two-mouse case and a physical prototype) to determine the effectiveness and affordances of the multi-touch technology for arranging information on a large-scale wall space in a paired collaborative task.

To assist in this study, a separate visualisation and interaction classification tool was developed, allowing the replay of XML log data in real time to assist in the video analysis required for observation and hypothesis testing.

CONTENTS

1. <i>Introduction</i>	12
1.1 Chapter Layout	16
1.2 Thesis Contributions	18
2. <i>Large-Scale Interactive Display</i>	19
2.1 Technologies for Interaction with Large-Scale Display	19
2.2 Developing a Large-Scale Interactive Display	22
2.2.1 Hardware	22
2.2.2 Software	27
2.2.3 Robustness and Latency	41
2.3 Emerging Interaction Technologies	48
2.4 Informal Case Study: Aichi	51
3. <i>Interaction Study</i>	55
3.1 Problem Domain/Related Work	55
3.2 Interaction Design Goals	58
3.3 Study Methodology	60
3.3.1 Task Description	61
3.3.2 Hypothesis Testing	63
3.3.3 Questionnaire Design	66

3.3.4	Study Procedure	69
3.3.5	Recruiting	72
3.4	Software	72
3.4.1	Architecture	73
3.4.2	Experiment Management	77
3.4.3	Supporting Multi-Mouse Interaction	78
4.	<i>Interaction Data Visualisation and Analysis Tool</i>	80
4.1	Problem Domain	80
4.2	Tool Requirements	81
4.3	XML Log Schema	82
4.4	Visualisation	84
4.5	Clutter Reduction	86
4.6	Data Browsing and Classification	87
4.7	Performance and Possible Improvements	89
5.	<i>Results</i>	93
5.1	Demographics	93
5.2	Commentary on Computer Vision System	94
5.3	Log Data Analysis	95
5.3.1	Hypothesis H_1	95
5.3.2	Hypothesis H_2	96
5.3.3	Hypothesis H_3	98
5.3.4	Hypothesis H_4	99
5.3.5	Hypothesis H_5	100
5.4	Questionnaire Data	102
5.4.1	Observations	110

6. <i>Conclusions and Future Work</i>	114
7. <i>Bibliography</i>	119
 <i>Appendix</i>	 128
A. <i>Questionnaire Forms</i>	129
B. <i>Study Result Tables</i>	139

LIST OF FIGURES

1.1	Muybridge Zoopraxiscope: late 19th Century disc-based image projector. [Source: Stuart Grais, DePaul University, Chicago]	13
2.1	IR-sensitive filters separate input and display elements on a large-scale interactive display. [Source: Sony CSL]	20
2.2	Cooperative work using i-Land's DynaWall. [Source: Fraunhofer IPSI]	21
2.3	NEC WT-600 projector mounted below and behind screen surface.	23
2.4	ADS webcam with attached IR filter plastic, mounted between four IR illumination panels. [Taken with Sony NightShot camera.] .	24
2.5	Initial LED sources did not provide even enough illumination over the surface.	25
2.6	LED strips used to determine optimal LED positioning.	26
2.7	Computer Vision application processing flow.	28
2.8	Interactions where forearms are partially detected are clipped to the most likely hand area.	38
2.9	Tracking blobs. Concentric circles show positions of blob centroids in subsequent frames. Note that an interaction on the left-hand side of the screen area has just finished.	40

2.10	Uneven illumination across the surface makes robust hand shape detection difficult. [Taken with Sony NightShot camera.]	42
2.11	hueview utility visualises luminance as colour.	43
2.12	Calibration using checkerboard. [Source: OpenCV Manual]	45
2.13	Shared Design Space interactive screen, featuring DNP Holo- Screen material and a twin camera configuration. [Source: FH- Hagenberg OoT]	49
2.14	Finger detection through frustrated total internal reflection (FTIR) [Source: NYU MRL]	50
2.15	Interactive screen in use at World Expo 2005. [Source: Story! Inc]	51
2.16	Setting up multiple touch screens at Aichi.	52
2.17	German interns testing interaction performance in New Zealand.	53
3.1	Matched physical cards placed on screen display, with matched virtual cards for comparison.	61
3.2	Card configurations for tasks $\{T_A \dots T_D\}$	63
3.3	Two USB mice used in the study for multi-mouse condition.	70
3.4	Early Memory screenshot featuring cards in different states of ac- tivity. Note slight card transparency.	74
3.5	Multi-mouse method. Cursors are colour-coded to differentiate between different users.	76
3.6	BoundedHand cursor features a yellow circle for interaction, plus a translucent “shadow” representing the bounding box.	76
3.7	ExperimentMenu handles study logging/recording duties.	77
4.1	Visualisation tool early in development.	85
4.2	ControlPanel in detail.	88

4.3	A number of “worst case scenario” interactions on a single-touch screen log.	90
4.4	Data from a noisy screen interaction.	91
5.1	Good hand recognition	94
5.2	Bad recognition due to uneven illumination	95
5.3	Marginal recognition due to tilted hand posture	96
5.4	H_1 : Time taken to complete task	97
5.5	H_2 : Ratio of participant interaction time	97
5.6	H_3 : $Sum(d1)/Sum(d2)$ efficiency measure	99
5.7	H_4 : Normalised pair matching times	100
5.8	H_5 : Comparison of turn-taking behaviour between input methods	101
5.9	Participant usage of wall spaces and computers	102
5.10	Questionnaire Results Summary — (Q_{01} – Q_{07})	103
5.11	Questionnaire Results Summary — (Q_{08} – Q_{13})	105
5.12	Questionnaire Results Summary — (Q_{14} – Q_{20})	107
5.13	Preference ranking of four methods	109

LIST OF TABLES

3.1	Experimental Design	62
5.1	Ranking interface methods by preference	110
B.1	H_1 : T_x means grouped by method.	139
B.2	H_2 : R_x means grouped by method.	139
B.3	H_3 : $Sum(d1)/Sum(d2)$ means grouped by method.	139
B.4	Q_{01} : “I found the task requirements easy to understand”	140
B.5	Q_{02} : “I found the task easy to achieve”	140
B.6	Q_{03} : “I did most of the work in performing the task”	140
B.7	Q_{04} : “I was distracted by what my partner was doing”	141
B.8	Q_{05} : “My input was disrupted by what my partner was doing” .	141
B.9	Q_{06} : “It was easy to use this interface with a partner”	141
B.10	Q_{07} : “It would be easier to use this interface by myself”	142
B.11	Q_{08} : “I feel that I paid attention to what my partner was doing”	142
B.12	Q_{09} : “I feel that my partner paid attention to me”	143
B.13	Q_{10} : “I feel that I provided useful input to my partner”	143
B.14	Q_{11} : “I feel that my partner provided useful input”	144
B.15	Q_{12} : “I feel that I could learn from observing my partner’s actions”	144
B.16	Q_{13} : “I feel that I could demonstrate things to my partner” . . .	144
B.17	Q_{14} : “The application was stable”	145

B.18	Q_{15} : “The application behaved predictably”	145
B.19	Q_{16} : “It was easy to move cards around the task space”	145
B.20	Q_{17} : “It was easy to correct any mistakes that were made” . . .	146
B.21	Q_{18} : “It was easy to recover from any errors in the interface” . .	146
B.22	Q_{19} : “This method provides a good way to arrange information”	146
B.23	Q_{20} : “This method felt like a natural way to solve the task” . . .	147

1. INTRODUCTION

Large surfaces have been employed for the display of information for a long time. Spaces such as bulletin boards and blackboards (and more recently whiteboards) provide a way for people to display and organise large amounts of information, often for a large number of people. Traditionally, these have been passive media — all interaction with the information is human-initiated.

With the advent of projection technology (from the zoetropes and “magic lanterns” of the 19th Century through to modern video projectors) we gained the ability to project active information into such spaces, but information so displayed could not be manipulated *in situ* by those referring to it. Modern electronics and computing technology allow users the opportunity to interact more directly with their information.

While a desktop computer is a good platform for interacting privately with information in a personal space, this is not the only domain in which computer technology can be employed. Large-scale interactive displays do not directly supplant traditional desktop computing; rather, they provide a reactive medium that augments the existing physical surfaces in which groups of people communicate [21].

This thesis builds upon prior research dealing with interactive surfaces, primarily employing Jun Rekimoto’s HoloWall[18] technology. To further specify the problem domain, the research looks at interactions that are:

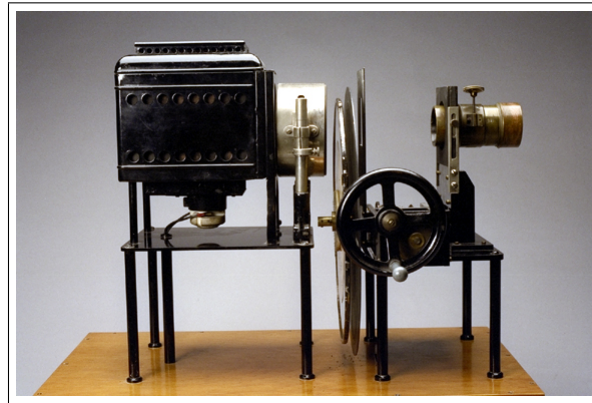


Fig. 1.1: Muybridge Zoopraxiscope: late 19th Century disc-based image projector.
[Source: Stuart Grais, DePaul University, Chicago]

- **Using vertically-aligned spaces.** Vertical information space is often used in different ways from horizontally-aligned information space [44], even when the information being displayed is very similar. A map room may contain vertical maps that provide overviews and persistent information, while a map table is used to coordinate movements in a more dynamic fashion.
- **Large scale.** Whereas smaller-scale displays such as a desktop monitor, laptop or PDA provide information in a form suitable for a small audience (typically one or two people) in close physical proximity, large-scale displays are generally intended to provide information to a larger audience, over a larger physical area. Larger areas also allow more cohesive understanding of the relationships between large collections of information without losing spatial context and peripheral awareness [17].
- **Multi user.** Having multiple users interacting with and within a shared information space introduces issues of concurrency (simultaneous action

versus turn-taking), information ownership and privacy [33, 31, 14].

- **Face-to-face collaborative efforts.** Remote collaboration through methods such as teleconferencing and groupware can be performed using existing single-user devices, and thus have different requirements in terms of interaction design. This study draws on the principles of Single Display Groupware (SDG) [33, 34], which suggests that the physically proximate users with a shared display encourages more natural and effective communication than remote collaboration or the use of separate displays [31].
- **Multi touch.** Allowing multiple, concurrent interactions with information space using the same physical device can effect changes in the way that groups interact (especially in terms of turn-taking) [28, 20]. It can also increase the amount of control bandwidth available to individual users [2], allowing two separate interactions or bi-manual methods for input and control.
- **Performed in a public or semi-public space.** In a private space, users have the opportunity to customise a device to their own particular needs or behaviours. While wall spaces in a single-person office space may have a primarily private role, they can be used in a semi-public role when information needs to be shared or discussed with visitors [21]. In a group workspace, wall spaces can be used semi-publically for internal coordination and visualisation, and in a more public fashion to display information to people external to the group [19].

In a public or semi-public space, there should be allowances made for many individuals to interact with an information appliance. This can be done by

tracking individual users, or providing a shared interface that is designed to support *user agnostic* interaction without any user-specific policies [20]. Additionally the number, role, level of involvement and grouping of participants involved in collaboration may change rapidly over the course of their activities.

- **Tool free.** There are some kinds of interaction in which it makes sense to use a certain kind of tool. For example, sketching or annotation tasks are generally easiest to do with some kind of pen or stylus. However, other interactions such as arranging or sharing content may be more naturally performed using bare hands [26], or a combination of bare-handed and tool-using interactions [1]. Reliance on tools can also restrict the number of users that can interact concurrently, affecting their turn-taking behaviours. Another design problem that needs to be addressed is that if users are presented with too many physical tool choices with different affordances, they can become confused and distracted [17].
- **Supporting an existing design practice.** While this research initially looked at a broader problem domain, there is a dearth of fundamental research pertaining to this relatively new kind of interface. Considering that in Gaines' BRETAM model of development [7, 6], most of the research is still taking place at the Breakthrough or Replication end of the spectrum, getting at least some Empirical data became a goal. This also led to a tighter focus: studying the process of arranging items of information on the screen surface. After getting concepts into the design space during a brainstorming phase, it is common to then cluster items together in a conceptually-related fashion [17, 22]. This process is sometimes referred

to as “affinity diagramming” [4].

In modern-day offices, teams of people may have a “war room”, or space where they can collaborate to generate, collect, discuss and form strategies for dealing with large pools of information [22]. Having only a single individual able to interact with such a space at once often means that all discussion is mediated through the attention span of one individual, or that participants must take turns making their contributions. In either case, valuable ideas may be lost that could be captured if all participants can collaborate and interact with such a discussion space concurrently [12]. Open collaboration on design artifacts can also engender a sense of collective ownership and engagement that is not present to the same extent in mediated collaboration, where the design artifacts are seen as being “owned” primarily by the mediator [17].

When developing an information appliance, it is important to consider the context in which it will be used. The potential users, their typical (and possibly atypical) tasks and physical space affordances must be considered. It could also be of benefit to consider other information appliances that may exist in these spaces (the user’s personal devices, physically co-located devices and task-related devices regardless of their location). A number of research initiatives into ubiquitous computing [36, 39, 23, 30] suggest the advantages of having several information appliances (such as tables, wall spaces and PDAs) amongst which data and design artifacts can be shared and transferred.

1.1 Chapter Layout

The layout of this thesis is intended to first introduce broad topics and prior research and technology to the reader, providing material to underpin the main

focus of the work. This is split into two major focus areas: the interaction technology, and a comparative user study to determine the advantages and disadvantages that this technology provides.

In **Chapter 2**, large-scale interactive screen technology will be described, followed by a detailing of the computer vision solution employed for this research.

In **Chapter 3**, the focus will be on the properties of large-scale screens, how they have been used for interaction and collaboration in the past, and the goals for and design of the interaction study undertaken.

Chapter 4 outlines the tool developed in order to assist in classification and visualisation of user study log data in combination with video footage, as an adjunct to Chapter 3.

Chapter 5 presents the results and subsequent analysis of the study. This chapter includes the results of hypothesis testing, a review of questionnaire data and other observations made during the study. In this chapter the multi-touch screen is directly compared with other collaborative interface methods: using multiple mice, a single touch screen or a physical prototype with no computer support whatsoever.

Chapter 6 provides a conclusion detailing the goals achieved and knowledge gained from this research, and suggests possible further directions which may be explored.

The thesis is concluded with a bibliography, and the information sheets and questionnaire forms used in the interaction study are included as appendices, along with more comprehensive study results tables.

1.2 Thesis Contributions

This thesis makes a number of contributions to research in this field. A lightweight multi-point tracking system provides processor-efficient responsiveness to a computer vision based system. A tool has been created for the efficient classification of interaction data based on video analysis. A large trial has been conducted of the technology in a public space, with the associated hurdles of ensuring the quality and reliability of both software and hardware.

The most important individual contribution within the thesis is the *direct* comparison between the multi-touch screen interaction and other methods that can either explicitly or implicitly support multiple users' tasks in a large scale wall/screen space. Even though it is only a pilot study, the results gained can inform future research of some of the advantages, drawbacks, do-s and don't-s of conducting this kind of research, and highlights the challenges that need to be faced in order to make multi-touch screen technology not just a human interface, but a truly *humane* interface.

2. LARGE-SCALE INTERACTIVE DISPLAY

This chapter details the technology and context in which the multi-touch screen prototype was developed, and the hardware and software considerations involved in that development. Given that this research began as a part-time course of study and later included a period of severe illness that lasted several months, technologies that have emerged during and after this work are detailed later in Section 2.3. To conclude the chapter, an informal case study is presented dealing with the use of this multi-touch display technology at the New Zealand pavilion at World Expo 2005 in Aichi, Japan — perhaps one of the largest trials of a new multi-touch interface of its kind in the world to date.

2.1 *Technologies for Interaction with Large-Scale Display*

This project builds primarily on Jun Rekimoto’s *Holo Wall* [18, 25] project (see Fig. 2.1). The display projection surface is illuminated with infrared (IR) light. Normally the light diffuses into the environment, but when an object such as a hand is placed close to or onto the surface, the light is reflected back behind the screen where an IR-filtered camera can sense it.

One of the advantages of this approach is that objects that are close to the screen but not touching will still be detectable as dim, diffused shapes while touching objects will appear as crisp, non-diffused images. Given ideal illumi-

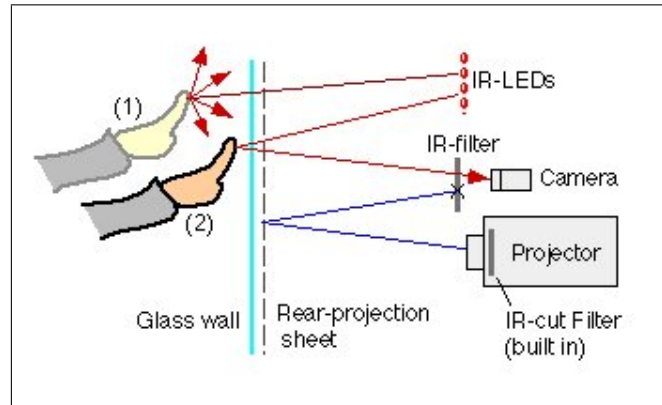


Fig. 2.1: IR-sensitive filters separate input and display elements on a large-scale interactive display. [Source: Sony CSL]

nation, this would allow us to employ thresholding techniques to differentiate between a “hover” and a “touch” interaction. A vision-based approach also allows us multiple concurrent interaction points on a single interactive pane.

The touch-sensitive *DynaWall* used in the AMBIENTE *i-Land* project [36, 39] at the Fraunhofer Institut employs a number of stroke-based gesture methods [8] for moving objects around a large display space. This project identified that it is difficult to employ some traditional keyboard- and mouse-based techniques; for instance, to “drag-and-drop” an object from one side of the *DynaWall* to the other would require you to keep your hand in constant contact with the surface over a distance of around four metres. The *DynaWall* is also limited by its underlying technology; the SMARTBoard used can only detect one interaction at a time; two or more concurrent touches will appear to the system as one single touch at the average of their positions.

SMART Technologies have since developed the Digital Vision Touch (DViT) screen that uses corner-mounted cameras in the screen bezel to detect multi-



Fig. 2.2: Cooperative work using i-Land's DynaWall. [Source: Fraunhofer IPSI]

ple points of touch¹. New Zealand-based firm NextWindow is using a similar technique in their touch screen technology². One drawback with this technique is that shapes may occlude each other from the viewpoint of one or more of the cameras, making the triangulation of multiple interactions difficult. On a larger screen that could theoretically support larger numbers of interactions, the chance of this causing manipulation problems for the users would increase considerably.

The *Barehands* technique [26] developed at Stanford and Brown Universities uses a hybrid approach, with back-projected touch-sensitive SMARTBoard technology used to prompt a computer vision system for classifying very simple hand poses (horizontal edge, vertical edge, one finger, two fingers, palm, unknown).

They also introduce the concept of an *overface* — providing traditional (e.g. mouse-like) interactions to existing applications while providing a wider range of techniques for interaction with enhanced applications, or to manage the display

¹ <http://www.smarttech.com/dvit/index.asp> [2006]

² <http://www.nextwindow.com/products/index.html> [2006]

itself.

Other technologies have been used to interact with large-scale display spaces. Many of these require the use of additional hardware, such as specialised magnetic trackers to record the position of a tool that is used for the interaction [3]. However, this technology is intended to allow people to use a human method of interaction first and foremost [41] — their bare hands. The concept of “walk-up usability” is a priority in a multi-user setting, especially when interactive displays are to be used in public or semi-public places.

2.2 Developing a Large-Scale Interactive Display

The section describes the creation of the interactive display, the methodology employed, and discusses the various challenges and setbacks faced in implementing the computer vision solution, both in software and hardware.

Given the goal of supporting existing design practices, the prototype was build to take up a relatively small space, taking up a depth of just over half a metre. Several applications of large rear-projected screen technology in the past [36, 26] have required the use of a separate room behind a wall area in order to project sufficiently large images; in designing a large-scale interactive device, it is important to take into account the *architectural* affordances of the design space, as well as considering what tasks those spaces are used for.

2.2.1 Hardware

The prototype hardware developed for experimental testing uses a high-end workstation with a Pentium processor (A dual-core 2.8Ghz Pentium 4), to increase the performance of the Intel OpenCV³ computer vision library used in

³ <http://www.intel.com/technology/computing/opencv/index.htm> [2006]

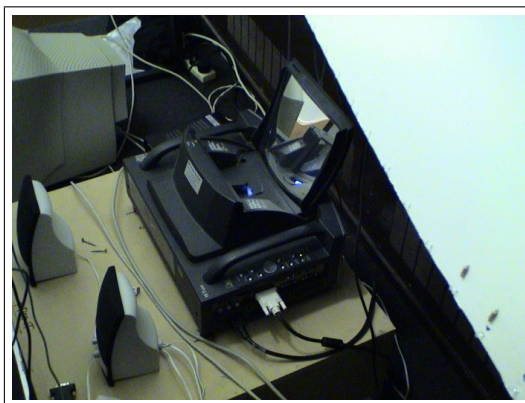


Fig. 2.3: NEC WT-600 projector mounted below and behind screen surface.

software development. This library is optimised for use on their Pentium processors. The screen surface itself is a large piece of glass (for a durable, cleanable surface), supported by a metal framework. The ADS⁴ USB 2.0 camera and illumination elements are mounted on a board approximately 500mm behind the screen surface. The camera's lens was replaced with a wide field-of-vision lens so that the camera can be mounted close behind the screen surface. Given the small amount of space behind the screen, an NEC WT-600⁵ "wide throw" projector is used. The projector uses an array of specially-shaped mirrors within the unit to emit a large image without requiring the projector to sit a long distance behind the screen, thus reducing the architectural form factor of the technology considerably.

Taking into account the 4:3 aspect ratio of the camera's CMOS sensor element, measurement of the screen determined dimensions of approximately 1600mm by 1200mm (64" by 48") were available for interaction. To determine the angle of lens necessary to view the entire screen surface, some simple

⁴ USB2.0 Turbo WebCam (USBX2020): <http://www.adstech.com/> [2006]

⁵ <http://www.nec-pj.com/products/wt/index.html> [2006]

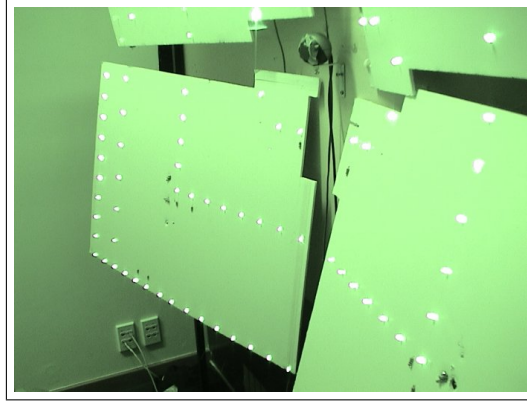


Fig. 2.4: ADS webcam with attached IR filter plastic, mounted between four IR illumination panels. [Taken with Sony NightShot camera.]

trigonometric calculations were used. Constructing a triangle using half the width and depth of the prototype:

$$\begin{aligned}
 \tan(.5x) &= opp/adj \\
 .5x &= \tan^{-1}(800mm/500mm) \\
 &\approx 58^\circ \\
 x &\approx 116^\circ
 \end{aligned} \tag{2.1}$$

Taking into account approximations in calculation and available lens sizes, a 130° lens was trialled, and found to be a suitable size.

The CMOS image sensor inside the webcam picks up IR light by default; indeed, many webcam lens assemblies include an IR-cut filter by default, so that any IR light in the environment does not interfere with a regular visual-spectrum image. By removing the default lens assembly and replacing it with a new one, it was possible to ensure that no IR-cut filter was used, allowing a lower overall level of IR illumination to be used. A filter that cuts visible light was attached over the front of the webcam (see Fig. 2.4).

As discussed in Rekimoto’s *SmartSkin* and *HoloWall* work [24, 18, 25], sensitivity to lighting conditions was a significant hurdle in building the prototype — care had to be taken to get an even illumination level across the surface⁶, and to remove strong external sources of infrared light (e.g. sunlight) from the environment.

Initial experiments used arrays of densely-packed 15° half-angle IR LEDs (see Fig. 2.5), but these were found to cause too much of a “spotlight” effect upon the screen surface, and so are not useful for achieving an even level of illumination. Sparse arrays of wider-angle (30°) LEDs were constructed in strips in an attempt to determine the optimal pattern for illumination of the screen (see Fig. 12.6), which led to the current hardware: using four panels of LEDs on angled gimbal mounts, to provide a more even level of illumination across the screen surface. Even then, however, the panels and even individual LEDs had to be adjusted carefully to attain the requisite uniform level of illumination).



Fig. 2.5: Initial LED sources did not provide even enough illumination over the surface.

⁶ see § 2.2.3 for further discussion of the calibration process.

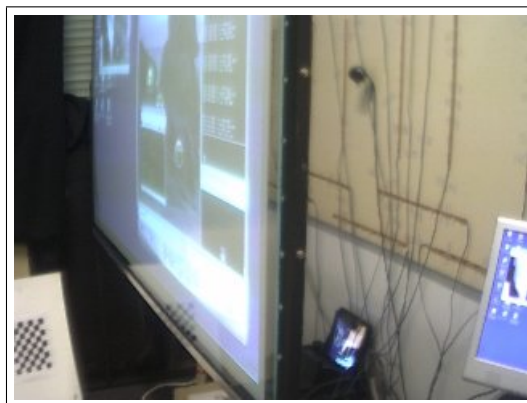


Fig. 2.6: LED strips used to determine optimal LED positioning.

Screen Materials

It was found early on that synthetic dyes and textiles (such as nylon and polyester) tend to be quite IR-reflective, so a cotton voile (both bleached and unbleached) was trialled. It was found to reduce the display light levels a small amount, was prone to dirt and creasing, and degraded image quality. Diffusion was also weak enough that there were bright IR element reflections visible from the external protective glass sheet.

Two varieties of specially-textured diffusing plastic were tested, varying in the fineness of the texture. Both were found to produce good diffusion for CV purposes, but ran into issues as a display surface. The light from the floor-mounted projector intersects the screen on quite a sharp angle, and the nature of the material meant that more light left the surface on that incident angle than on any other angle; the image was quite bright when viewed from 60° above the screen surface, dim when viewed directly at eye level, and virtually invisible from 60° below the screen.

Currently, the back of the protective glass screen is coated with a thin layer

of “frosted vinyl” (produced and applied onto the glass by a local manufacturer), which provides a decent degree of diffusion, good display image quality, and low IR reflection properties. Given that the degree of diffusion that was desired was difficult to quantify, a number of sheets of frosted vinyl were applied iteratively on the surface until a sufficiently diffusing substrate had been created (using four sheets). Ideally, a single sheet of the appropriate thickness would provide a better solution, with crisper projected image clarity and less chance of small blemishes appearing on the surface due to imperfections in the layer application process.

2.2.2 *Software*

The process of converting the camera image into output suitable for a user interface is an involved process. On one hand, the computer vision system must address the physical model of the hardware being used to capture the interaction — properties of the camera and lens assembly, screen position and materials, and projected image registration. On the other hand, it must address the requirements of the user in terms of what kind of interactions are desirable and intentional, and present those interactions in a form that provides usable information to a user interface system.

After initialisation and systems checks, image frames are retrieved from the camera and pre-processed to address camera-specific issues such as image noise and extraneous colour depth. The images are then processed to extract interesting image features, producing a number of discrete binarised contours (a.k.a. “blobs”) where there is contact with the screen. These blobs are then further classified and filtered to extract the meaningful interactions (or part of an interaction). The collections of blobs are then tracked from frame to frame and

classified into discrete interactions. Interactions are then passed through the network export interface into one of a number of external applications that may handle the user interface part of the system.

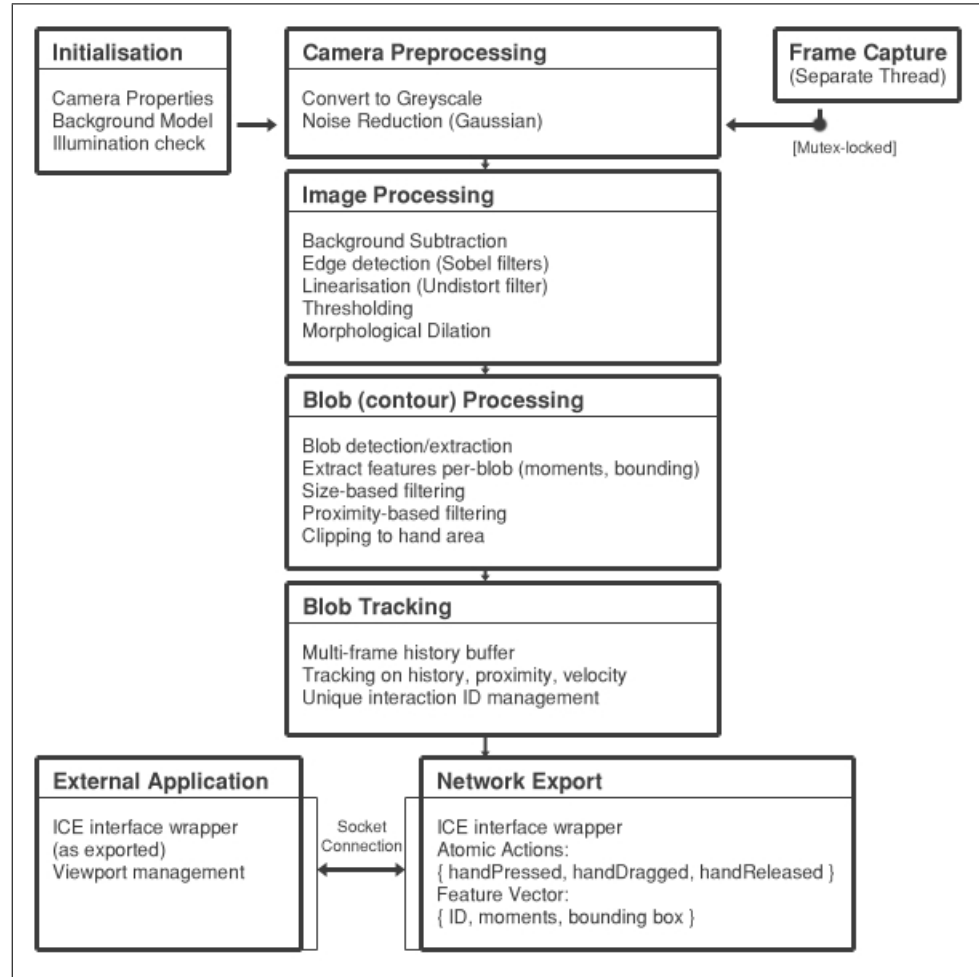


Fig. 2.7: Computer Vision application processing flow.

As Fig. 2.7 shows, the design is intended to be extensible, modularised so that component parts can be swapped out in favour of another approach, or even branched so that different subsystems can extract different kinds of information

that may be collected together later (e.g. for hand pose classification [32, 26]).

Initialisation

The CV system starts by loading in resource files containing calibration data such as camera lens characteristics, image thresholding values and the tolerances used in parts of the code that check for hardware faults.

When the application is started, the first few frames captured are used to build a model of the background image used later in background subtraction.

One approach to background subtraction that was attempted (but not eventually used) to solve the problem of uneven lighting was “painting” a maximal image: wiping a white cloth across the screen surface to build a map of maximum input levels on a per-pixel basis. This (pre-generated) static maximal image (**lightmap**) and a regular minimal background image (**darkmap**) would form the theoretical upper and lower bounds for input, allowing us to capture interaction better. This approach, while initially promising, proved unstable over time as small changes in lighting conditions caused by movement of the screen prototype could require a new maximal image to be made.

In this stage, various image buffers used in later processing are also initialised, along with kernel matrices used for 2-D convolution algorithms. Since most of the operations require a fixed amount of memory, it is considerably more efficient to allocate the required space once, rather than continually reallocating it from frame to frame.

Camera Capture and Pre-processing

The camera captures image frames at low resolution (320x240), but a reasonable speed (30 frames per second). Since the IR light captured has little colour dif-

ferentiation on a CMOS sensor, the image is converted directly into a greyscale image to reduce the processing overhead.

Given the low levels of light used, the image frames captured by the camera can be quite noisy. A simple 3x3 Gaussian smoothing filter is applied at this point to reduce this effect. Note that while this does blur the useful parts of image somewhat as well; reduction of noise in this early step reduces the processor overhead that it would cause in later contour detection. A 3x3 sharpening filter is applied directly after smoothing to bring the remaining detail into stronger definition.

Image Processing

The initial approach used for background subtraction normalised an incoming frame with the pre-calibrated maximal and minimal images⁷ (`lightmap` and `darkmap`):

$$I_{Norm} = \frac{(I_{Input} - I_{Min})}{(I_{Max} - I_{Min})} \quad (2.2)$$

Given the physical instability of the metal prototype framework and its illumination panels, this technique proved less than robust in practice: movement of the screen hardware could cause a rapid change of lighting levels any given pixel, even to the point of creating a light level far lower than the “minimal” image. This technique might prove more useful when used in conjunction with a more rigid screen enclosure.

A new approach was devised that generated an “edge map” as the back-

⁷ This technique was devised concurrently with and independently of Andrew Wilson’s TouchLight work [43] in which a very similar technique was published, demonstrating parallel evolution to solve similar needs.

ground model of the system. An image is constructed by applying Sobel edge detection filters (3x3 convolution kernels to detect edges in each horizontal, vertical or diagonal direction) to the minimal image and adding the results to each other. This results in a background edge-image that is less susceptible to small physical movements than the earlier approach.

$$I_{MinEdgeMap} = \frac{1}{N} \cdot \sum_{n \in N} \sum_{K_s \in K} filter(I_{Input_n}, K_s) \quad (2.3)$$

where K is the set of Sobel edge detection kernels and N is the number of frames of background used in calibration. The general convolution formula is as follows:

$$filter(I, K, p) := \sum_{i \in R} v_{i,p} \cdot k_i \quad (2.4)$$

where R is the (3x3) region of convolution, adjusted for image array boundaries, k_i is the value of cell i in kernel K and $v_{i,p}$ is the corresponding value in the sliding window around each image pixel $p \in I$.

Incoming images are similarly convoluted prior to background subtraction, yielding a much cleaner image showing only the edges of objects in close proximity to the screen:

$$I_{Sub} = I_{Norm} - I_{MinEdgeMap} \quad (2.5)$$

It is important to ensure that features detected in the camera frame can be mapped directly onto the tangible space in which the interactions take place, so that a user can reach out and touch the part of the screen he or she wishes to interact with directly. If hardware and design considerations allowed for a

large space behind the screen, a camera lens and a narrow field of vision could be combined with an ad-hoc translation method to yield acceptable results.

However, due to the form factor requirements in this system, the camera must be placed only a short distance behind the screen. This requires the use of a wide-angled lens that creates a significant “fisheye effect” distortion in the image frame. In order to correct for this distortion, certain intrinsic parameters must be calculated for the camera-lens assembly⁸. The OpenCV library contains routines to calculate these intrinsic values in a calibration process [46, 47] that will be discussed in more detail in §2.2.3.

Assuming a common rotational and translatory frame of reference, and disregarding for a moment the issue of radial distortion, the relationship between a true point $p = (x, y, z)$ and its observed position $\tilde{p} = (\tilde{x}, \tilde{y}, \tilde{z})$ is given as:

$$p = A^{-1}\tilde{p} \quad (2.6)$$

where A is a matrix containing intrinsic parameters (c_x, c_y) , the coordinates of the principal point of focus, and (f_x, f_y) , specifying focal lengths in terms of the X and Y axes:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Radial distortion is characterised by four calculated coefficients k_1, k_2, p_1, p_2 .

These coefficients describe a polynomial⁹ such that:

⁸ Since the camera is in a fixed position relative to the screen, we do not need to account for extrinsic (rotation and translation) parameters when mapping between camera and screen coordinates.

⁹ OpenCV Manual, pg. 6-2.

$$\tilde{x} = x + x[k_1r^2 + k_2r^4] + [2p_1xy + p_2(r^2 + 2x^2)] \quad (2.8)$$

$$\tilde{y} = y + y[k_1r^2 + k_2r^4] + [2p_1xy + p_2(r^2 + 2y^2)] \quad (2.9)$$

$$r = \sqrt{x^2 + y^2} \quad (2.10)$$

The intrinsic matrix and coefficients are used in the OpenCV `cvUndistortInit()` to generate an array of mappings between each observed camera pixel and its corrected position. This array is then used to linearise incoming images:

$$I_{Linear} = \text{cvUndistort}(I_{Sub}) \quad (2.11)$$

After being linearised, the resultant image is masked (to avoid picking up the screen hardware visible at the periphery of the image) and thresholded at an optimal value T , turning it into a binary image¹⁰ suitable for contour detection:

$$I_{Thresh}(x, y) = \begin{cases} 1 & \text{if } ((I_{Linear}(x, y) > T) \\ & \wedge (I_{Linear}(x, y) \in I_{PeripheralMask})) \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

Note that the threshold value T must be calibrated with care: too high a value would result in the system missing deliberate actions, while too low a setting runs the risk of letting through false positive actions, where a user's silhouette moving behind the screen may inadvertently trigger an interaction.

The image is then subject to morphological operations. Areas where a con-

¹⁰ Note that this assumes an image binarised with values in the set $\{0, 1\}$; in an 8-bit OpenCV binary thresholded image, this 1 actually refers to a value of 255. $\{0, 1\}$ values are used in these formulae for illustrative simplicity.

tour may have been broken by the thresholding process can be regenerated by morphological dilation of the image. A subsequent erosion operation is performed to maintain initial contour boundaries and smooth off some of the “rough” parts of binarised edge contours, reducing the noisiness of the image in preparation for blob detection.

Erosion and dilation are essentially set operations performed on a binarised image. A structuring element is iterated over the image, and pixels are included in or excluded from the resultant image based on the interaction between the image and the structuring element. An arbitrary 3x3 structuring element with a central origin/anchor point was used, allowing regions to be grown or reduced evenly in all directions.

A basic formula shared by each operation is translation of an image A by a pixel p :

$$A_p = a + p | a \in A \quad (2.13)$$

For a dilation operation, a new image pixel is set to 1 if *any* part of the overlaid structuring element B is positioned over a pixel from image A that is also set to 1:

$$A \oplus B = \bigcup_{b_i \in B} A_{b_i} \quad (2.14)$$

For an erosion operation, a new image pixel is set to 1 if *all* parts of the overlaid structuring element B are positioned over pixels from image A that are set to 1:

$$A \ominus B = \{p | B_p \subseteq A\} \quad (2.15)$$

These operations are performed sequentially on the source thresholded image, dilating and then eroding twice each:

$$I_{Morph} = (((I_{Thresh} \oplus B) \oplus B) \ominus B) \ominus B \quad (2.16)$$

Blob Detection and Processing

There are several methods in the OpenCV library for finding blob contours in a binary image. The blobs that need to be detected are hand outlines that should not lie within the boundaries of any other contours. A hand edge outline may have “holes” (zero-value areas) within the shape that an algorithm could detect, but they do not provide any extra relevant information in this case¹¹. As such, the simplest algorithm was employed: detecting only the external contours without looking for any holes within those blobs (or, recursively, further sub-blobs within those holes) [37].

Contours are initially stored using an 8-connected directional chain code before being converted to point values. To reduce complexity and storage size, a contiguous segment of contour pixels with the same chain code value (horizontal/vertical/diagonal direction) are compressed, with only the start and end points of the run being stored.

At this point a useful piece of information is also calculated for each blob: the bounding box. The size of this bounding rectangle is used to approximate

¹¹ Note however that such information could prove useful in a more advanced algorithm capable of detecting where two hand outlines touched each other, and possibly splitting the contour appropriately.

the size of the hand shape, and perform some initial filtering to reduce the amount of data processed in later steps; a blob that has a bounding box size that is considerably larger or smaller than that of a whole-hand interaction is discarded.

Blobs then undergo further feature classification and identification. Each contour is drawn onto a scratch image and flood-filled. These images are then used as the basis for moment analysis:

$$m_{pq} = \sum_{x \in X} \sum_{y \in Y} x^p y^q . B(x, y) \quad (2.17)$$

where X and Y span the range of (x, y) values in the bounding box of a shape, and B is a binary map of the shape area (i.e. 1 where the shape exists, 0 otherwise).

The 0^{th} -order moment m_{00} gives the non-approximated area of the shape. 1^{st} -order moments m_{01} and m_{10} are used to calculate the centre of mass (centroid) of a shape. 2^{nd} -order moments m_{02} , m_{11} , m_{20} are used to find the principal axis of the shape¹².

$$A = m_{00}, \quad \bar{x} = \frac{m_{01}}{m_{00}}, \quad \bar{y} = \frac{m_{10}}{m_{00}} \quad (2.18)$$

Blob feature sets are then stored in a vector associated with each frame.

In some cases a hand contour may not be well detected, resulting in two smaller blobs that lie close together. As each new blob is added to the vector, it is compared against existing blobs in that image for proximity. The distance between the centres of mass is calculated:

¹² Rotational information is not used in the interaction study tasks detailed later, but it would be a valuable feature for hand pose classification algorithms if richer pose/gesture interaction classification methods were to be implemented.

$$dist(a, b) := \sqrt{|\bar{x}_a - \bar{x}_b|^2 + |\bar{y}_a - \bar{y}_b|^2} \quad (2.19)$$

If the blobs are deemed to be too close to each other, they are considered to be the same interaction, their bounding boxes are added and their features combined:

$$A_{New} = A_a + A_b \quad (2.20)$$

$$p = \frac{A_a}{A_{New}} \quad (2.21)$$

$$\bar{x}_{New} = p.\bar{x}_a + (1 - p).\bar{x}_b \quad (2.22)$$

$$\bar{y}_{New} = p.\bar{y}_a + (1 - p).\bar{y}_b \quad (2.23)$$

When touching the screen, it is possible that the CV system may pick up part of the wrist and forearm as well as the hand area. To minimise this effect, blobs that are larger than hand-sized (but still small enough to be considered a valid interaction) are clipped, reducing the region of interest to the most likely hand area. As the palm and fingers of the hand generally occupy a larger area than the wrist and forearm, that part of the region will contain more mass and thus attract the centroid (\bar{x}, \bar{y}) toward it. The relationship between the centroid and the central point of the bounding box can be used to determine which part of the image is most likely to be the hand, and the bounding box is clipped appropriately (Fig. 2.8).

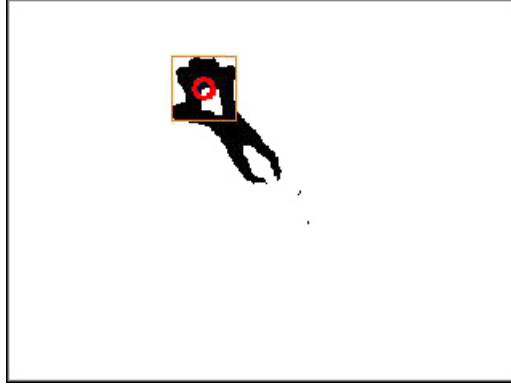


Fig. 2.8: Interactions where forearms are partially detected are clipped to the most likely hand area.

Blob Tracking

Once processed in each frame, blob vectors are tracked over multiple frames, separating the collections of shapes out into individual interactions as users' hands move across the screen space. Human limb movement is assumed to maintain a relatively constant velocity from frame to frame, with no discontinuities of motion (i.e. no abrupt changes in direction).

As interactions are tracked, velocity information can be built up to more accurately estimate the positions where new blobs may occur, improving the reliability of tracking. More advanced tracking techniques such as Kalman and Condensation [15] filters are available for this purpose, but a simpler tracking algorithm was found to track multiple hands satisfactorily with a very low processor overhead to enable tracking at full frame rate and maintain interface responsiveness.

The algorithm used to track blobs from frame to frame is as follows:

- Let M be the set of blobs in the frame being considered

- Let N be the set of blobs in the previous frame
- If no blobs are in frame N , assign new IDs to all blobs in M
- Construct $|M| \times |N|$ distance matrix $d[i, j]$ (using formula 2.19)
- Construct choice mapping array, so that each blob in M is mapped to the nearest blob in N :

$$\forall i \in M, \text{choose}[i] := \min(d[i, j]) \forall j \in N \quad (2.24)$$

- Where two blobs map to the same blob in the previous frame, mark the choice that is farther away as invalid:

$$\text{invalid}[i] := (\exists i_2 \in M) \wedge (\text{choose}[i] = \text{choose}[i_2]) \wedge (d(i, j) > d(i_2, j)) \quad (2.25)$$

- Invalidated choices are treated as the start of new interactions
- Valid choices are assigned the ID of the blob they mapped to in the previous frame
- Any blobs remaining unchosen from the previous frame are considered to have finished if they have not been seen after a certain number of frames (SKIP_FRAME_THRESH). Otherwise, their positions are updated according to their previously-calculated velocities.

Network Export

Once blob movements have been tracked and classified into interaction elements, their information can be exported to another application (or applications) via

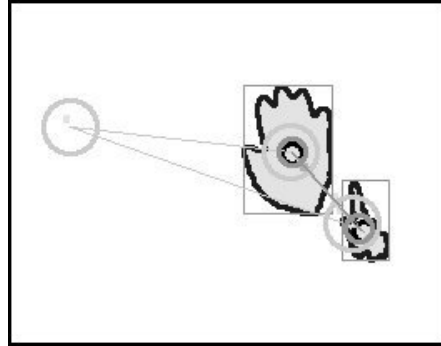


Fig. 2.9: Tracking blobs. Concentric circles show positions of blob centroids in subsequent frames. Note that an interaction on the left-hand side of the screen area has just finished.

network socket. Each atomic network action identifies the specific interaction ID associated with the blob, an ID associated with the screen device, and appropriate blob features.

When a new interaction is started, a `handPressed()` action is initiated. When an interaction is considered finished, a `handReleased()` action is initiated. Between these two actions, as a blob is being tracked and updated from frame to frame, a `handDragged()` action is initiated.

Other housekeeping actions performed by the network export class include:

- `registerDevice()`: obtains a device ID when the CV system initialises
- `unregisterDevice()`: releases the device ID when the CV system shuts down
- `registerViewport()`: sends screen boundary information to the target application, so that coordinate systems can be remapped appropriately
- `reportError()`: reports any hardware-related errors that are detected to the target application (such as LED panel failure)

- `testConnection()`: tests the connection so that the application can be closed down automatically when the connection is lost, if such behaviour is desirable

The process of prototyping network communications was sped up dramatically by the use of the Internet Connection Engine (ICE) library by ZeroC Inc.¹³. Its Slice interface specification language can compile into network code for a number of languages on a number of platforms, making it suitable for a wide range of inter-process communication applications. This allowed rapid changes to the network interface without having to sacrifice robust communication and spend hours on debugging and testing refactored code.

Development experiences with ICE indicate that it could be a good basis for constructing a wider suite of ubiquitous computing applications and architecture, similar to the SmallTalk-based BEACH architecture [38] used in Fraunhofer's AMBIENTE project or, in another domain, the CORBA-based Distributed Wearable Augmented Reality Framework (DWARF) developed at TU München¹⁴.

2.2.3 Robustness and Latency

An information appliance should ideally provide a *reliable* (fault-tolerant) and *responsive* (low latency) interface to users [17]. In order to promote that reliability and responsiveness, a number of factors needed to be addressed.

With the prototype hardware and test environment available, this calibration process was quite difficult and time-consuming to complete. A number of small tools were written and employed to make the process flow more smoothly, but

¹³ <http://www.zeroc.com/ice.html> [2006]

¹⁴ <http://ar.in.tum.de/Chair/ProjectDwarf> [2006]

calibration issues remain an important problem to address in building a reliable and responsive interface.

Illumination calibration

Providing a constant level of illumination (needed for reliable blob detection) proved to be a more involved process than initially envisioned. Even after moving away from the IR “spotlight” panels to the use of four large angled panels, adjusting those panels and the LEDs on them to provide even lighting levels across the surface takes a considerable amount of time. During the course of this project such adjustments had to be performed several times due to the hardware being moved to accommodate other research projects in the lab, or dismantled and used for other purposes some time into a period of extended absence due to illness.

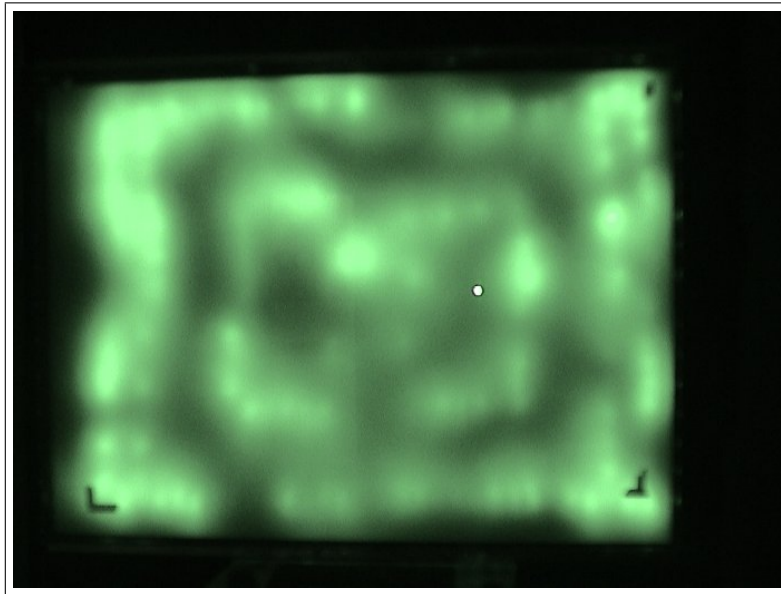


Fig. 2.10: Uneven illumination across the surface makes robust hand shape detection difficult. [Taken with Sony NightShot camera.]

One of the challenges involved in this process is the nature of local contrast perception in the human visual system. Our eyes are far better at detecting large changes in illumination levels than fine gradients, and we find it difficult to precisely identify the absolute lighting level of a given region¹⁵. This problem is compounded when using a low contrast projector to view the image (though a CRT monitor was sometimes brought in to help for this purpose).

The `hueview` utility captures a camera image and provides two different images: a light map that shows luminance values and a hue map that converts those luminance values into a colour spectrum. This makes it easier for us to perceive which areas are lit at a certain level. By ensuring that as much of the screen as possible was within the same colour band, it is possible to minimise the number of areas where CV system sensitivity was low.

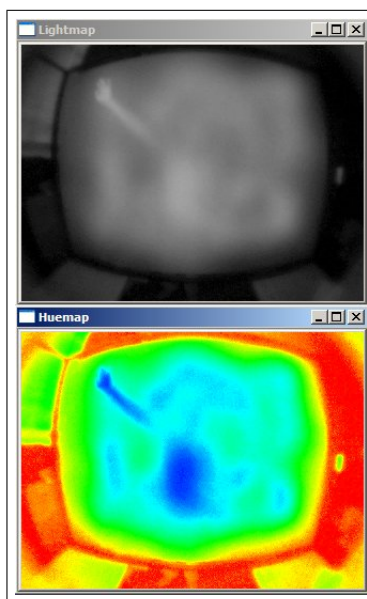


Fig. 2.11: `hueview` utility visualises luminance as colour.

¹⁵ See http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html [2006] for an example and discussion of this effect.

Image registration

The process of image registration involves a number of different coordinate spaces. Physical points on a screen are translated into image frame coordinates by the camera assembly, translated into virtual screen coordinates by the computer vision system and then back into physical coordinates by the projector and its positioning in the world. Calibration of various parts of the system are necessary so that the projected physical screen coordinates match the initial physical positioning of the screen.

In other interaction methods, a user's input and output spaces are often offset from each other (e.g. moving a mouse on a table to move a screen cursor). The degree of separation between these spaces has an effect on the performance of users, and when the spaces are drastically offset the user can suffer a considerable drop in task performance [42]. However, in a case where visual and kinesthetic feedback is given, the visual channel tends to dominate [1] feedback and a user can compensate by adjusting their kinesthetic actions.

If the interface performs in a consistent fashion it is possible for a user to learn to compensate over time, even when the disjunction is an extreme one [35], however this learning period can cause considerable frustration and disorientation. As such, having precise image registration is a usability priority for information appliances.

Each camera's sensor-lens assembly may have different intrinsic characteristics: we cannot rely on assuming that the centre of the lens is placed directly over the top of the centre of the sensor array, and each physical lens may have slightly different optical characteristics within certain tolerances. These characteristics must be measured in order to achieve precise image registration.

Fortunately, the OpenCV library distribution contains not only the algorithms for performing calibration [46, 47], but an implementation in the form of a DirectX filter (**CalibFilter**). Using the **GraphEdit** application included in the DirectX SDK, a small calibration tool was constructed.

To start with, it is important to ensure that the camera lens assembly is in focus, such that the entire screen surface is as clear as possible. A blurry image reduces the strength of edges used in the calibration process (and later during blob detection).

In the calibration process, a checkerboard pattern glued to a piece of stiff cardboard is held within the camera's field of view. As the number and size of the squares are known quantities, the filter can detect the checkerboard's position. Corner positions of each of the squares in the checkerboard are used to measure the observed size and shape of each square versus an ideal model of the board. Several samples are taken with the checkerboard pattern in different positions and orientations, from which the focal length and radial distortion characteristics can be estimated. Due to the large amount of radial distortion caused by the 130° lens employed, 20 samples were taken (instead of the filter's default of 10) to improve precision in the figures obtained.



Fig. 2.12: Calibration using checkerboard. [Source: OpenCV Manual]

Due to a bug in the driver code available for the camera, calibration could only be performed at 1280x960 resolution. However, the parameters obtained from that calibration were easily converted into those suitable for the 320x240 resolution used in the CV system by dividing them by a factor of 4 where appropriate.

Once the camera is calibrated against the screen, the extent of the linearised image space can be discovered (by interacting at the corner points of the screen), and the projector can be moved and/or digitally adjusted so that the projected image matches what is detected by the camera.

Hardware fault detection

As IR light is invisible to the naked eye, it might not be immediately apparent that some fault had occurred in the illumination panels. The CV system periodically checks illumination levels over each quadrant of the screen. If illumination falls below a certain threshold level, the system will start sending `reportError()` events so that the UI application can display or log errors, so that a technician could easily diagnose the problem and replace a faulty component. If the illumination level drop was temporary in nature, `reportError()` can send an “all clear” code, showing that the hardware is once more functioning normally.

Software pipeline

In code used early in development, fetching the image frame from the camera and processing that frame took place in the same thread. As processor load increased, however, increasing levels of latency became noticeable. The `cvQueryFrame()` function used in the OpenCV library took a long time to fetch

each frame from the ADS camera used.

To solve this issue, a different approach to camera capture was taken. The OpenCV function was swapped out on favour of the CodeVis VidCapture¹⁶ library, which provided the ability to constantly stream data from the ADS camera at its maximum frame rate of 30 frames per second without the extra delays caused by fetching frames one at a time.

Running the camera capture in a thread concurrent with the image processing thread allowed the CPU's dual-core architecture to be exploited, and provided an increase of around 10fps to an average of 28fps. The memory shared by the two threads was mutex-locked, to avoid problems that would be caused by reading a frame as another was being written from the camera.

Frames not used due to the image processing taking longer than $1/30^{th}$ of a second are discarded, so that the system is always processing the most recent image available, ensuring that the interaction does not lag drastically behind real movement.

Balancing reliability and responsiveness

When tracking objects from frame to frame it is possible that a blob may not be present for one or more frames, due to the user briefly moving their hand away from the touch surface or areas of low illumination sensitivity. If an interaction were to stop dead while a virtual object was being dragged across the interface, the user would then have to stop and reacquire that object. In order to account for this behaviour, the tracking algorithm uses velocity information to predict where any given blob is going to be in the next frame.

However, there is a problem with allowing this kind of predictive method

¹⁶ <http://www.codevis.com/vidcapture/> [2005]

to smooth over any holes in an interaction: what happens when you want to stop? When a user removes his or her hand from the screen, there may be a small movement in the X-Y plane prior to removal. This movement may be used to predict the position of that hand for a number of frames after the hand is removed, resulting in the virtual object being dragged moving to a different position from where the user intended before being released.

The `SKIP_FRAME_THRESH` value mentioned in § 2.2.2 determines the number of frames that pass before a blob is released by the tracking algorithm. If this value is too low, then the user may drop objects they are dragging when they don't intend to. If this value is too high, then objects may “fly away” when released.

To minimise the fly-away effect, a damping factor (`VELOCITY_DAMP_FACTOR`) was introduced, so that the predicted velocity of a blob was lessened with each passing frame that it was not detected. This damping does however detract from the reliability of the tracking predictions, so care should be taken not to dampen too much. After testing and balancing these factors against each other, a `SKIP_FRAME_THRESH` value of two frames and a `VELOCITY_DAMP_FACTOR` of 0.7 were used. The assumptions made about the nature of human movement were found in practice to be valid ones.

2.3 *Emerging Interaction Technologies*

Since initial development began on the CV system, a number of new interaction devices have been developed. The TouchLight [43] screen developed by Microsoft Research employs a special holographic film¹⁷ to project an image seemingly

¹⁷ DNP HoloScreen: <http://www.en.dnp.dk/get/472.html> [2006]

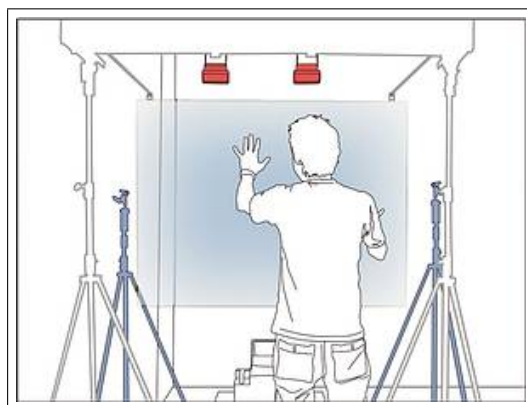


Fig. 2.13: Shared Design Space interactive screen, featuring DNP HoloScreen material and a twin camera configuration. [Source: FH-Hagenberg OoT]

into free space, while IR light is used to illuminate users in front of the screen without being disturbed by the visual image. Touch registration is performed by a fusion of images from stereo cameras behind the screen.

The “Office of Tomorrow”¹⁸ project at the Upper Austria University of Applied Sciences (FH-Hagenberg; a HIT Lab NZ virtual worlds consortium academic partner) has developed a 60-inch FlipTouch screen that combines the approach used in the CV system with that of TouchLight, and is incorporating the technology into their Shared Design Space project, showcased at SIGGRAPH 2006.

Another technology that has garnered a lot of attention and accolades in recent months is Jefferson Han’s work at the New York University Media Research Lab (MRL) on touch sensing through frustrated total internal reflection (FTIR) [10], where touching the screen surface causes diffusion of light that would normally not escape from the screen material due to its narrow angle of incidence with respect to the screen/air boundary. The screen surface used

¹⁸ <http://www.officeoftomorrow.org/> [2006]

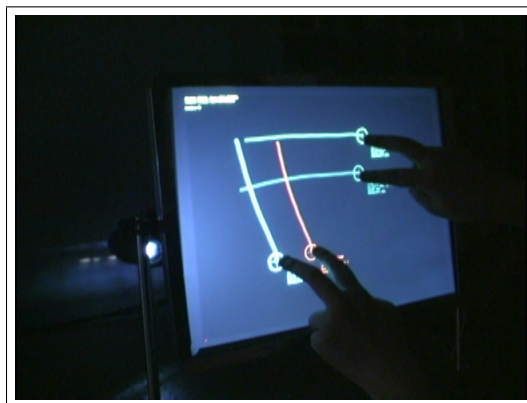


Fig. 2.14: Finger detection through frustrated total internal reflection (FTIR) [Source: NYU MRL]

is an acrylic, making it more prone to scratching and general wear and tear than a glass screen, but Han suggests using a “water white” (low in iron) glass formulation may also be a suitable screen material.

Water white glass has far higher optical clarity than regular glass, transmitting up to 98–99% of light through a pane of glass (versus approximately 90% for normal glass formulations, characterised by a greenish tint when viewed side-on). While Han’s technique works well in its current 16-inch by 12-inch form factor, scaling the device up may cause a decrease in responsiveness due to attenuation of the light through the medium: the more light that can be transmitted through the substance used, the better. Another issue is that the fragility of a certain thickness of glass increases relative to the size of the pane, and this may have an effect on the suitability of the current FTIR design as form factors increase.

A similar parallel implementation of the FTIR technique has been developed in Italy by the Natural Interaction IO Research Center group¹⁹, where it is being

¹⁹ <http://www.naturalinteraction.org/> [2006]

used in the TabulaMaps project to support map browsing using two-handed interactions.

2.4 Informal Case Study: Aichi

The interactive zone in the NZ Pavilion at World Expo 2005 in Aichi, Japan featured five large-scale (80-inch diagonal) screens, using a slightly earlier version of the computer vision system detailed here. Five screens were set up in a semi-circular configuration, each with background images combining to form panoramic views of New Zealand scenery. Content flowed from screen to screen (synchronised via a network interface) in the form of small circular pictures. These pictures could be touched and dragged down to target areas at the bottom of the screen. When moved near the targets, they would be locked into place and activate a video or slideshow animation showing content associated with that picture.

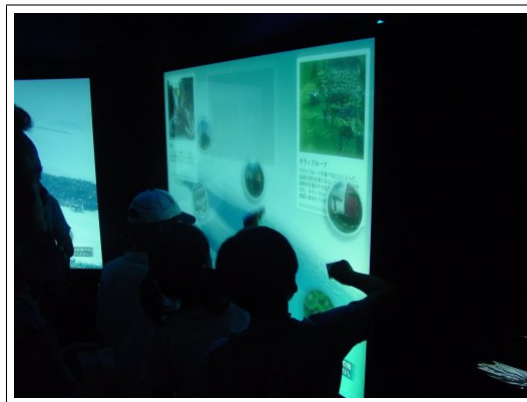


Fig. 2.15: Interactive screen in use at World Expo 2005. [Source: Story! Inc]

Originally the pictures were intended to be iconic in nature, but in the final presentation small photographic images were used to represent various areas

of life in New Zealand and its environment, such as “Sports and Adventure”, “The Four Seasons” and “Education”. The exhibit’s content was developed by exhibition designers Story! Inc. in collaboration with design company Oktobor for the New Zealand Ministry of Trade and Enterprise.

Developing the exhibit highlighted a number of the issues that needed to be addressed for developing a system as a public (or semi-public) information appliance. Physically, the materials used need to be able to withstand wear and tear over a long period, and be easy to clean and maintain. Any faults needed to be diagnosed and fixed within a short timespan to minimise downtime in the installation. The installation also took place in the middle of a Japanese summer; care had to be taken in the design to vent the exhaust heat caused by five projectors and six computers away so as not to cause any hardware faults from overheating, or environmental discomfort to the users.



Fig. 2.16: Setting up multiple touch screens at Aichi.

On the software side of things, the code had to run for approximately 10–12 hours a day, seven days a week for a period of six months without crashing. After extensive testing here in New Zealand, the software worked as expected

for the entirety of the six-month exhibition, with more than a third of the estimated four million visitors using the technology²⁰.

The testing process included some stress testing methods, such as leaving the application running for several days at a time, with thresholds set down at a level where environmental factors would occasionally be detected as interactions. By employing this method, any memory leaks or overflow conditions that occurred either within the software or associated libraries could be identified, then isolated and fixed over time.

When testing user interaction, care was taken to include people who were not immediately involved with the project, as what may have seemed intuitive for designers and developers may not been intuitive for the general public. This included a number of Asian students who were to some extent familiar with Japanese culture.

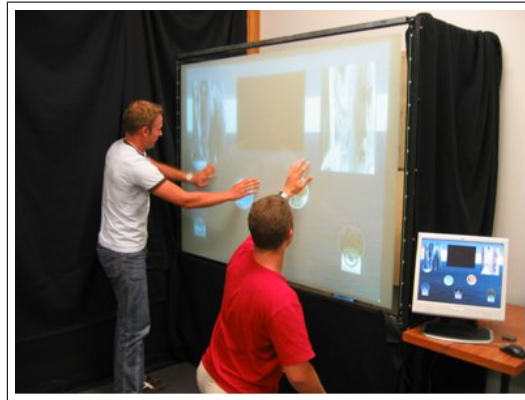


Fig. 2.17: German interns testing interaction performance in New Zealand.

The environment, number of users and cultural habits had an effect on the interaction method used. Though the computer vision system could handle

²⁰ Personal correspondence with James McLean, Story! Inc.

bare-handed interaction, it was considered undesirable to have hundreds of people with sweaty hands to be touching the screens directly each day, both for hygiene reasons and because the finger-marks might over time detract from picture clarity. Instead, each visitor was given a business card folded in half so that they could hold half of the card and press the other half against the screen to touch things, rather than interacting directly.

In terms of exhibit interaction design, feedback from staff on-site suggests that having content that was loosely structured and immediate in presentation was beneficial in creating an engaging experience. Other exhibitions had interactives that were ponderous by comparison; visitors often walked away rather than waiting for some action to complete and the system to return some degree of control to the user again. This illustrates the value of providing timely feedback and continued control in an information appliance: people who are required to wait for operations to complete may soon become impatient.

As well as enjoying the slideshow and video content, many visitors (especially children) also enjoyed the visual and tactile experience of simply moving pictures around on the screen.

3. INTERACTION STUDY

Large scale multi-touch screen interfaces are a relatively new invention. Quite aside from the novelty of new technology, its availability prompts an important question: “What is it good for?”

While there are a number of potential applications in such areas as interactive exhibition kiosks (§2.4) and art installations, this part of the study focuses on investigating a particular domain in which a number of devices and techniques are currently used — not all of them computer-supported.

Specifically, it will look at whether and to what extent the properties of the large-scale, multi-touch screen technology support the arranging of data in during collaborative design processes, in direct comparison with other manipulation methods.

In terms of the domain of information addressed, it will be assumed that the individual data items in question will have some degree of similarity, either in terms of a hierarchy or conceptual linking, as arrangement of relatively dissimilar data is a considerably more arbitrary task.

3.1 Problem Domain/Related Work

In a traditional office setting, walls are often used to support information spaces [19]: blackboards, whiteboards, notice boards, projection walls and the like. Given

that large amounts of information can be placed in such spaces, that information is often gathered into clumps of similar data [21], presumably to reduce the amount of time and effort that users must spend searching the space to find what they are looking for.

When people are collaborating on the design of something that consists of or pertains to a wide range of concepts, brainstorming is a commonly-used tool, allowing groups to get various individual ideas down into a common space, identify similarities and creating categories that the ideas may be grouped into, and making connections either within or between those categories. The process of gathering similar ideas into groups within the information space is commonly referred to as “affinity diagramming” [4, 17].

While this study focuses on the brainstorming design style, and affinity diagramming behaviour in particular, it is worth noting that this method is not the only approach: the Berkeley Group for User Interface Research (GUIR)’s “Designer’s Outpost” [16] study identified that when creating web sites, users who were visual designers usually started by creating a page prototype, but those who were primarily information architects preferred to start from a site map, spreading “breadth-first” into what pages would be needed in the site and how best to categorise them, and moving into specific page content and graphical design.

These observations informed their DENIM prototyping tool [22], which aims to support both techniques, using a sketching interface with different zoom levels to create low-fidelity website prototypes. This allows a user to create pages and manage the connections between them at a high zoom level, or create a page then zoom in to detail individual pages and functionality. This design can then

be compiled into a navigable website prototype. It should be noted, however, that this interface relied on a single-user tablet PC, rather than the vertical screen-based interaction of Designer’s Outpost.

Why should we use a computer solution at all? Information presented on the wall space can also be updated more dynamically to account for changes that are occurring in work elsewhere in the office or task environment. With a physical noticeboard, each datum must be explicitly placed on or removed from the space, and information posted there may fall out of date more quickly if someone doesn’t consider updating it to be a priority. Guimbretière’s FlowMenu [9] work at Stanford also suggests while designers like using simple techniques like scribbling out notes and sticking them up on a wall¹ when they are generating ideas, the process of collecting and transcribing those artifacts into a form that is usable after the brainstorming phase is distinctly less enjoyable. While devices such as whiteboards that can be “cycled” to scan and print their contents, or cameras may be used to create an image of a wall space, computer-supported interactive walls enable users to have a direct digital reproduction of the task space that may be transferred directly to other devices or domains where they continue the design and development process.

Another area of research that informs this study is Single Display Groupware (SDG) [33, 40]. Not limited to interactive wall spaces, SDG deals more generally with situations in which a display device is shared collaboratively between several users who are all capable of input. Each user may have his or her own input device (such as a mouse), or use a shared input device capable of detecting multiple users’ interactions, such as the MERL DiamondTouch ta-

¹ The Post-It[®] note seems to a very common design artifact mentioned in related literature.

ble² which uses capacitance circuits to differentiate between users³. While SDG suggests that users in the same physical space collaborate more naturally and effectively using a shared display, research suggests that “new conflicts and frustrations may arise between users when they attempt simultaneous incompatible actions” [33], so the nature of such conflicts should be addressed.

3.2 *Interaction Design Goals*

This study aims to further a number of goals in the problem domain, drawing on a number of studies into different phenomena of face-to-face collaboration with similar devices. These goals are:

- To determine how well multi-touch interaction (i.e. the ability to interact with two or more distinct areas at once) on large scale interactive displays models using a natural, physical device (e.g. a whiteboard or a bulletin board), to perform tasks such as brainstorming and affinity diagramming (placing conceptually-linked artifacts in close physical proximity, prior to formal structure elements being decided).

A physical method has tangibility affordances that touch screens do not. The ability to directly manipulate notes — to pick them up, hold them in one’s hand, and so forth — are missing from a touch screen. As such, interactions taking place on a screen may suffer from users being less able to see, move or share objects [16].

How “natural” a method is is a subjective matter requiring qualitative

² <http://www.merl.com/projects/DiamondTouch/> [2006]

³ This is a useful technique for when users are sitting around a table and signal antennae can be placed in seats to piggyback on the human electrical field — however, this technique for user detection may not be as directly applicable to people standing in front of an interactive screen, especially in areas where walk-up usability is deemed important.

assessment. Effectiveness in completing a task can be used as a more quantitative comparator between different methods.

- To determine how effective multi-touch interaction is in completing a set task compared to single-touch methods (i.e. where only one interaction at a time can be captured, or multiple points are averaged due to lack of ability to differentiate between interactions). Given the common use of single-touch devices such as the SMARTBoard in interactive wall research [36, 26, 16, 27], determining whether being able to perform multiple actions simultaneously significantly affects user efficiency.
- To observe whether different styles of interaction emerge (e.g. taking turns, one person dominates, etc.) depending on which interaction method is used. A number of studies have shown [28, 27, 17] that interaction is typically in either an *open* style, where all users contribute without a clear leader, or a *facilitator* style, where one user moderates discussion and input to the interface.

Another facet of communication where input methods may have interesting effects is that of turn-taking. In conversation, turn-taking is a social negotiation where participants can maintain or hand off the role of speaking to other participants. Given that some input methods can only be used by one user at a time, turn-taking applies not only to task collaboration, but collaborating in using the interface.

- To observe the effects of users' interpersonal space on their interactions. The NASA MERBoard [27] research observed that there are social rules governing the act of reaching across another user's personal space (an act considered rude or discomforting). Personal space issues are not as

prevalent in cases where each user has a separate input device and can reach “past” another user without invading personal space.

- To investigate how a multi-touch approach affects the degree of collaboration between partners. While this factor is somewhat subjective, relying on users’ impressions of how much collaboration they felt in their task, quantitative factors can also be captured, including patterns of turn-taking. Observational cues include speaking (either about the task at hand, or cues for turn-taking), and behaviour such as pointing.

3.3 *Study Methodology*

With these goals in mind, a pilot study was designed and undertaken to capture information that would allow these concepts to be observed and addressed. The multi-touch screen technology was compared against a number of competing input methods: a single-touch screen condition, a multiple mouse condition and a physical interaction condition. Users performed the study in groups of two. The study design was submitted to and approved by the University of Canterbury Human Ethics Committee.

As an abstraction of the conceptual clustering process, participants were required to perform a simple matching task, in which pairs of cards were placed on each interface and users were asked to match each pair. This encompasses the affinity diagramming concepts of gathering like items, and moving items around in the wall space.

Four different task variants were created, to counterbalance against the four input methods and so that a specific “set” of cards and their positions could not be learned from memory, thus introducing a learning effect into the study.

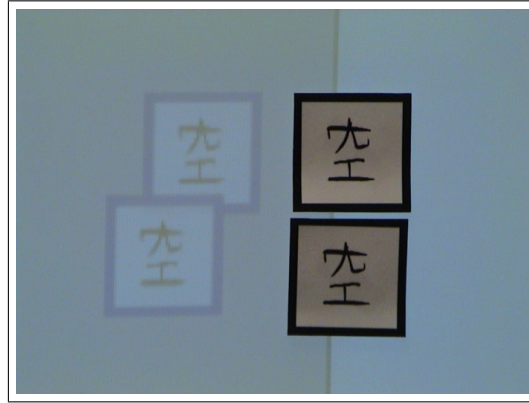


Fig. 3.1: Matched physical cards placed on screen display, with matched virtual cards for comparison.

Tab. 3.1 shows the experimental design consisting of: 4 tasks $\{T_A \dots T_D\} \times$ 4 methods multi-touch, single-touch, multi-mouse, physical interaction = 16 experiments, \times 2 participants per experiment = 32 participants overall. Cases are counterbalanced using a standard Latin square technique. Each interaction method will take place in the same physical screen space, to reduce bias between methods due to different display conditions. Due to contrast differences, physical interaction will take place with the room lights switched on.

3.3.1 Task Description

The each pair of cards featured a different simple Japanese *kanji* character, representing concepts such as “sun”, “book” or “water”. The characters were selected for their simple shapes, but to add a small degree of challenge to the task, each set contained some pairs of characters that differed only marginally. For example, the pictograms for “sun” and “moon” differ only in the length and shape of the two vertical strokes, and the characters for “water” and “wood”

Tab. 3.1: Experimental Design

Experiment	Participants	1	2	3	4
E_{01}	S_{01}, S_{02}	MT, T_B	ST, T_A	MM, T_C	PI, T_D
E_{02}	S_{03}, S_{04}	MT, T_C	ST, T_B	MM, T_D	PI, T_A
E_{03}	S_{05}, S_{06}	MT, T_D	ST, T_C	MM, T_A	PI, T_B
E_{04}	S_{07}, S_{08}	MT, T_A	ST, T_D	MM, T_B	PI, T_C
E_{05}	S_{09}, S_{10}	MM, T_B	MT, T_A	PI, T_C	ST, T_D
E_{06}	S_{11}, S_{12}	MM, T_C	MT, T_B	PI, T_D	ST, T_A
E_{07}	S_{13}, S_{14}	MM, T_D	MT, T_C	PI, T_A	ST, T_B
E_{08}	S_{15}, S_{16}	MM, T_A	MT, T_D	PI, T_B	ST, T_C
E_{09}	S_{17}, S_{18}	PI, T_B	MM, T_A	ST, T_C	MT, T_D
E_{10}	S_{19}, S_{20}	PI, T_C	MM, T_B	ST, T_D	MT, T_A
E_{11}	S_{21}, S_{22}	PI, T_D	MM, T_C	ST, T_A	MT, T_B
E_{12}	S_{23}, S_{24}	PI, T_A	MM, T_D	ST, T_B	MT, T_C
E_{13}	S_{25}, S_{26}	ST, T_B	PI, T_A	MT, T_C	MM, T_D
E_{14}	S_{27}, S_{28}	ST, T_C	PI, T_B	MT, T_D	MM, T_A
E_{15}	S_{29}, S_{30}	ST, T_D	PI, T_C	MT, T_A	MM, T_B
E_{16}	S_{31}, S_{32}	ST, T_A	PI, T_D	MT, T_B	MM, T_C

MT: Multi-touch, ST: Single-touch, MM: Multi-mouse, PI: Physical Interaction

are drawn very similarly indeed.⁴

Cards were placed in two columns on each side of the screen, with card pairs arranged so that some would be within each column and some would span both columns. This was to simulate users having both shared and individual tasks to complete on the shared interface.

In order to get some measure of the efficiency of moving cards across the surface in each of the three computer-supported cases, participants were also required to match cards together in a central space. The space was made wide enough relative to the cards and screen so that users could arrange pairs wherever they wished but did not necessarily have to move into each others' personal space to perform a match.

Physical cards consisted of printed characters glued to thin cardboard for

⁴ On the other hand, the use of *kanji* may, in hindsight, have introduced a linguistic bias into the difficulty of completing the task, and more abstract shapes should perhaps have been used.

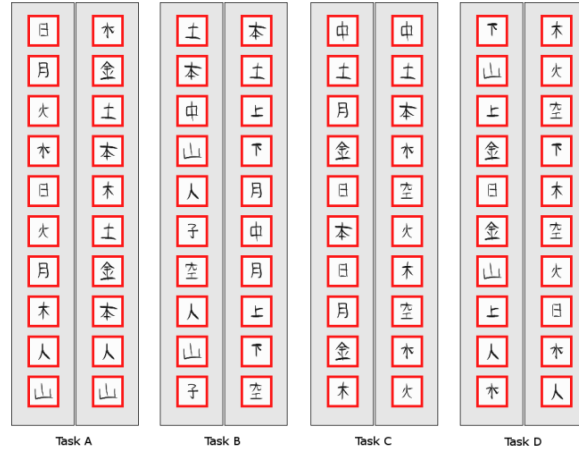


Fig. 3.2: Card configurations for tasks $\{T_A \dots T_D\}$.

extra rigidity and durability, and had the adhesive strip from a Post-It[®] note affixed to the back of the surface. Stronger adhesives were not used due to leaving a sticky residue on the glass surface which could affect both usability and optical clarity.

3.3.2 Hypothesis Testing

A number of hypotheses were formulated to capture user behaviours and task efficiency based on information that could be captured within the interface. The physical interface is only represented in H_1 , due to the difficulty of tracking physical cards both on and off the interface:

- H_1 : *There are no significant differences in the mean times taken to complete the task using different interface methods.*

This gives us an idea of how effective the participants were in achieving the task for each method, irrespective of the approach they took.

Measure: The time taken to match every pair of cards (T_x)

- H_2 : *There are no significant differences in the proportion of time that each participant spends interacting with the interface using each method.*

Measure: The ratio of time each participant $\{P_1, P_2\}$ spends interacting.

Note that as either participant could potentially dominate, and their ratios are complementary ($t(P_1) = 1 - t(P_2)$), the participant that interacted least is chosen, giving a possible result range of $[0, 0.5]$:

$$R_x = \frac{\min(t(P_1), t(P_2))}{t(P_1) + t(P_2)}$$

- H_3 : *There are no significant differences in the average distance each card was moved using each interface method.*

This is a measure of efficiency of interaction; also, it tells us about whether each card drag was as efficient as any other, or whether there was a lot of variation.

Note that because there was no specific fixed “target” for card movement other than the large central area of the position of another card, especially when two cards could be active and being moved at the same time, a Fitts’ Law measure [5] was not used. More in-depth analysis of log data, identifying cases where a card was obviously being dragged towards another motionless card, could still be possible in future, though a different task would probably be more appropriate for such a study.

If we do accept an (admittedly weak) assumption that cards are moved with the intent of reaching their final matched resting place, we could possibly use the ratio of distance travelled to the distance between a card’s starting and ending point as an approximate measure of task efficiency.

Measures: The mean total distance travelled by each card ($Sum(d1)$), and the mean total distance between a card's starting and ending points ($Sum(d2)$).

- H_4 : *There are no significant differences in the amount of time it takes to match pairs in (LL, LR, RR) configurations between each interface method.*

This may allow us to see if people are more likely to complete private tasks before tasks that may require more collaboration, and also if there are any differences in terms of sides of the screen that may be caused by factors such as the handedness of participants.

Measures: Mean times taken to match $\{LL, LR, RR\}$ pairs, normalised against $H_1 : T_x$.

- H_5 : *There are no significant differences in the amount of turn-taking that occurs between each interface method.*

This is difficult to quantify in terms of a single null hypothesis. Turn-taking is usually characterised by behaviour where only one person acts or speaks at once for most of the time [29]. In a multi-touch or multi-device case (or in the single-touch case when two users try to interact at the same time), this is not something that can be taken for granted.

The method taken for capturing participation is as follows.

- Construct a graph of states I, A, B, C .

I : represents an idle state

A : represents the state where only participant P_1 is acting

B : represents the state where only participant P_2 is acting

C : represents the state when both P_1 and P_2 are acting

Time spent in each of these states is measured as a baseline statistic. Over the course of a log, construct a string of state transitions. Skipping idle states, sharing behaviour is characterised by the following substrings:

- **AA or BB**: *Maintain* interaction (user stops, then continues)
- **AB or BA**: *Handoff* interaction (one user stops, another starts; classical turn-taking)
- **ACB or BCA**: *Interrupt* interaction (one user stops when another user starts; common in classical turn-taking but brief)
- **ACA or BCB**: *Interject* interaction (one user starts and stops while the other user continues)

It may be possible to compare the proportions of these events happening in various methods, and see if any interesting patterns arise.

3.3.3 Questionnaire Design

To capture a range of biographical and qualitative data about the experiment, participants were asked to fill out questionnaires (see Appendix A). An initial pre-experiment questionnaire gathered basic biographical information (age, occupation course of study, height, handedness), and gauged how often participants use desktop computers and wall spaces.

After each task was completed, participants were issued with the post-experiment questionnaire. This questionnaire presented 20 statements about the experiment and their paired collaboration. Participants were asked to rate each statement using a 7-point Likert scale.

As a design consideration, to differentiate the array of 140 rating circles, a key was included at the top of the page (see pg. 136), vertical lines were drawn on each side of the “Undecided” column and questions were grouped into three separate blocks with separate themes. The statements are as follows:

1. *“I found the task requirements easy to understand”*

A lack of understanding of task requirements may affect performance in carrying out the task, or a participants needs in terms of collaboration. In cases where task requirements are not clear for some reason, participants sharing an interface may still learn by observation, or by having a partner demonstrate what is required. If the study design procedure is complete enough, this statement should have a high level of agreement: users should not be confused by what is required of them, but we should know if they are.

2. *“I found the task easy to achieve”*

Hypothesis H_1 measures how effective participant pairs were using the time taken; this gives a subjective difficulty measurement for each participant.

3. *“I did most of the work in performing the task”*

This statement speaks to hypothesis H_2 , dealing with interaction styles and whether one user dominates interaction.

4. *“I was distracted by what my partner was doing”*

5. *“My input was disrupted by what my partner was doing”*

6. *“It was easy to use this interface with a partner”*

-
7. *"It would be easier to use this interface by myself"*

These statements look at the extent to which sharing the interface affects a user's interaction with the screen, and how much of their frustration was due to their partner's actions, versus difficulty with the interface itself.

8. *"I feel that I paid attention to what my partner was doing"*
9. *"I feel that my partner paid attention to me"*
10. *"I feel that I provided useful input to my partner"*
11. *"I feel that my partner provided useful input"*
12. *"I feel that I could learn from observing my partners actions"*
13. *"I feel that I could demonstrate things to my partner"*

This block of statements addresses participants' feelings of and about collaboration with each other: to extent to which attention was spent on the partner versus concentrating solely on the task, the level of contribution felt, and the ease of conveying things by interacting with the interface.

14. *"The application was stable"*
15. *"The application behaved predictably"*
16. *"It was easy to move cards around the task space"*
17. *"It was easy to correct any mistakes that were made"*
18. *"It was easy to recover from any errors in the interface"*
19. *"This method provides a good way to arrange information"*

20. *“This method felt like a natural way to solve the task”*

This block of statements looks at the stability and suitability of the interface method to the task. Results may allow us to identify particular areas of the interface design that require more attention.

On the reverse side of the post-task questionnaire, participants were asked for short answers to three questions, asking if there were any particular problems they found in using methods, whether could suggest any ways in which using the method could be improved for them, and if they had any further comments to make. This “short answer” section allows participants to reflect upon the experience in a more open style, informs future design processes and gives feedback on issues that may not be immediately obvious from statement rankings, data logs and video footage.

After the four tasks were completed, a post-experiment questionnaire asked participants to rate which methods they preferred by assigning a unique number from 1 (best) to 4 (worst) for each case. No differentiation was made between whether the methods were preferred for effectiveness or enjoyment, though users were given space to comment afterwards, and could give reasons for why they rated interfaces in the order they did.

3.3.4 Study Procedure

The following is a list of procedural guidelines used to plan the user study experiments:

- Participants will be paired to work together during the experiment.
- Participants will be seated at a desk facing away from the interface while

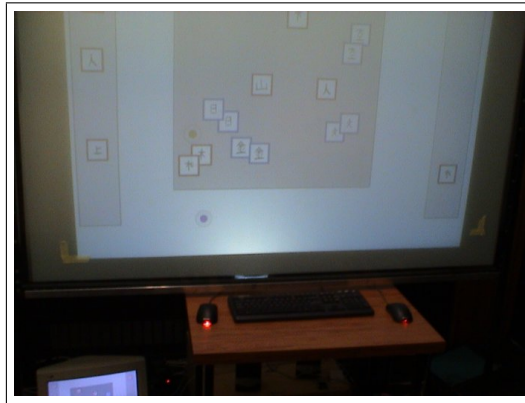


Fig. 3.3: Two USB mice used in the study for multi-mouse condition.

completing questionnaire forms. Spare pens should be provided, just in case.

- Before the experiment, each participant will fill in the pre-experiment questionnaire.
- In cases where handedness differs between participants, they should be seated such that their dominant hand is toward the inside of the pairing when they are facing the side of the screen that is closest to their seating position, to reduce the potential for bias from handedness.
- The multi-mouse method will be supported by placing a narrow table under the screen, providing a ledge to support the two mice.
- Participants may be given 5 minutes in which to become acquainted with each other, if they are not already. This may reduce the potential for bias in collaborating with unfamiliar people.
- The task will be explained, and each interaction method will be explained and demonstrated where applicable. Participants will perform a practice

session before live data is recorded, so that complete unfamiliarity with an interface or the task does not unfairly bias results. (No practice session for the physical case, as it takes too long to put all the cards back for the next task.)

- In touch screen cases, participants will be directed to move items with the palm or the back of the hand flat against the glass (for maximum responsiveness from the CV system). They will also be trained to move their hands directly away from the surface when releasing a card, to avoid cards “dropping” down the screen when a user drops his or her hand.
- Participants will be videotaped during the task, to pick up speech patterns and gestures for observational purposes, and to verify when and where each participant was interacting with the interface (c.f. computer-captured data).
- Data is to be kept private, secure and anonymised. Permission must be sought before any video footage or specific quotes from footage are used.
- Each task should take less than five minutes to complete. Including questionnaires and practice runs, the experiment must take less than an hour. Less than 50 minutes is probably desirable, akin to UC one-hour lecture times.
- After each task is completed, participants will be issued a post-task questionnaire pertaining to the interaction method they just used.
- After all interaction methods tasks have been completed, participants will be issued a post-experiment questionnaire sheet asking them to rank by preference (from 1-4) each of the interfaces they have used.

- An informal interview will be conducted after all questionnaires are returned, to gain insights or comments that may not have been written in the questionnaires.

3.3.5 Recruiting

Participants sought for the experiment were above the age of 18, of either gender and with no restrictions on height. No departmental budget was available to supply participant incentives for an HCI study, so a number of small bags of snack-sized chocolate bars were bought. The study was primarily advertised in the university student magazine (Canta), as well as through departmental email, personal weblog and word of mouth.

3.4 Software

In order to support the study, the **Memory**⁵ application needed to be developed to (a) allow users to carry out the task and (b) capture as much necessary information to support observation and hypothesis testing as possible using (c) all of the computer-supported input methods being assessed. A description of the task implementation follows:

The users are presented with a screen containing two vertical strips on either side of a central space. Each of these strips contains a number of rectangular cards featuring symbols. Each symbol appears on two cards, and these cards must be matched together in a central space. Cards are shown with a red border when they are unmatched, a green border when they are being actively manipulated and with a blue border when they are matched with another card. When cards

⁵ The application was named in honour of the game where players turn over pairs of cards in turn until they find all the matching pairs.

are matched together, they are locked together on the interface and cannot be moved. (This was a design choice made due to the possibility of accidentally picking up and “unmatching” cards in certain interaction methods. Ideally, it would have been preferable to then be able to move pairs as a single piece.) *The task is ended when all pairs have been matched.*

This description was used as a basis for introducing computer-supported methods to study participants.

3.4.1 Architecture

Some tools have been developed to aid in this kind of endeavour, such as the Single Display Groupware Toolkit [40] from the University of Calgary GroupLab⁶, and the Multiple Input Devices (MID) Java package from the University of Maryland HCIL⁷. Limitations in either the hardware, operating system or developer knowledge available caused an alternate approach to be chosen.

This software used in this study is based around Maryland HCIL’s Piccolo Zoomable User Interface (ZUI) Toolkit⁸. Piccolo allows developers to rapidly construct 2D structured graphics applications which can be zoomed into or out of (though this zooming functionality was not appropriate to the task). It is available in both C# and Java forms, though the Java version was used due to familiarity with the language. Aside from zooming, Piccolo also supports more general user interface functions such as bounds management, transparency, node movement and canvas layering.

Visual objects are represented as nodes in an object hierarchy, based on a generic **PNode** class that can be specialised into different kinds of objects (e.g.

⁶ <http://grouplab.cpsc.ucalgary.ca/> [2006]

⁷ <http://www.cs.umd.edu/hcil/mid/> [2006]

⁸ <http://www.cs.umd.edu/hcil/piccolo/> [2006]

`PImage` for images, `PText` for text strings and `PPath` for generic shapes such as rectangles or ellipses).

Like many user interface applications, the `Memory` application is based around a Model–View–Controller architectural model⁹. Model information for each card is stored in a `MemoryCard` class. This includes an unique ID, methods determine which other card(s) it can be matched with, and whether it is actively paired with another card. The `CardHolder` class extends from `PComposite`, and deals with presentation details relating to each card. Each state (unmatched, active, matched) is stored as a separate `PImage` child object, and display of each is handled through the `setVisible()` method on each. Card images were made slightly (10%) transparent, so that it was more obvious when cards were placed on top of one another (something that was immediately obvious in the physical case) [45].

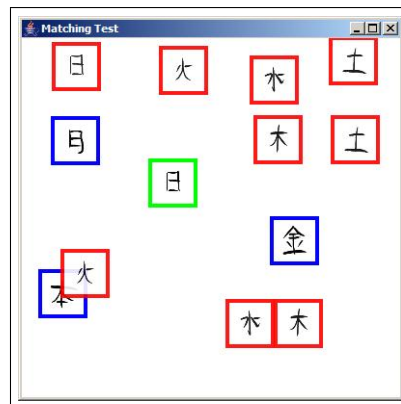


Fig. 3.4: Early `Memory` screenshot featuring cards in different states of activity. Note slight card transparency.

The `MemoryFrame` class (extended from `PFrame`) is the main window of the application. The `MemoryFrame` supports three separate layers: the *main layer*

⁹ <http://java.sun.com/blueprints/patterns/MVC-detailed.html> [2006]

which contains region rectangles and **CardHolder** objects, the *drag layer* which contains **CardHolder** objects that are currently being moved across the screen and the *cursor layer* where the cursor widgets are displayed. This layering allows the application to separate out graphical nodes that cannot be directly selected by new input.

Each **CardHolder** also has an associated **MatchHandler** object which acts as an intermediary for input that involves any given card. The **MatchHandler** keeps track of which interaction it is associated with, using a $(device, interaction)$ ID tuple (q.v. §2.2.2). It is also responsible for handling moving cards around the canvas (moving them up onto the drag layer when active, clipping card positions at viewport edges so that they don't inadvertently "fall off", etc.) and logging the distance travelled by the card (H_3).

The entry point for the **Memory** application is one of four executable classes, one for each input method. Each initialises whichever parts of the application are appropriate for its input. **IceServerScreen** and **IceServerSingleScreen** create an ICE socket server that listens for hand interactions through the **ScreenWrapperI** interface, managed through either a **ScreenInputManager** or **ScreenSingleInputManager**. **IceServerMouse** listens for mouse interactions through the **MouseWrapperI** interface, managed through a **MouseInputManager**. **IceServerPhysical** does not actually set up a network interface (see §3.3.2), but still handles setting up the display and experiment management components just as the other servers do.

Specific method input managers are responsible for displaying cursors on the interface. Multi-touch screen interactions use a **BoundedHand** widget (see Fig. 3.6), where a yellow circle sits atop a translucent grey shadow representing

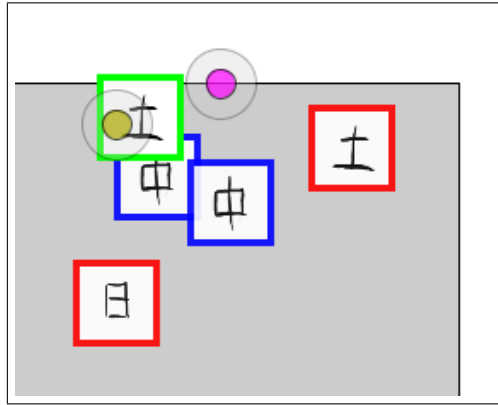


Fig. 3.5: Multi-mouse method. Cursors are colour-coded to differentiate between different users.

the bounding box of the shape. Single-touch screen interactions use a **Hand** widget, with only the yellow circle (as bounding box information is not available on single-touch devices). Mouse screen interactions use a **MultiCursor** widget, the same size as the circles used in the other widgets, but colour-coded (yellow or magenta) so that each user can better identify which cursor is theirs.

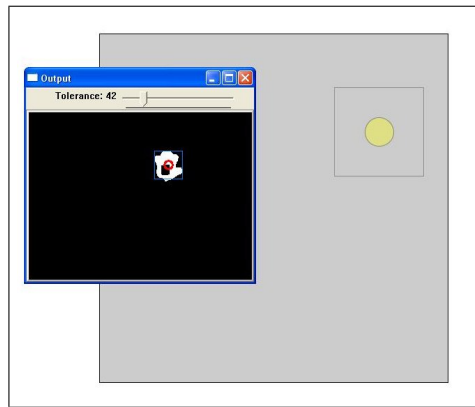


Fig. 3.6: **BoundedHand** cursor features a yellow circle for interaction, plus a translucent “shadow” representing the bounding box.

Each specific method input manager then passes input information into

GeneralInputManager, which handles interaction with the canvas. When an interaction starts, the **GeneralInputManager** translates its position into normalised screen viewport coordinates and determines what lies on the base layer of the canvas at that point. If a card is found, its **MatchHandler** is then associated with the interaction until it is released or a match occurs. When a match occurs, the cards involved are locked down on the base layer and a **match** event is logged by **XMLLog**¹⁰. This object is also responsible for logging **press**, **drag** and **release** events as they are dispatched by the **GeneralInputManager**.

3.4.2 Experiment Management

Experiment management functions are performed separately from the main interface in the **ExperimentMenu** window (Fig. 3.7). This small window contains controls that allow the researcher to specify a log ID in the text area, choose which of the four tasks is to be used, and start and stop the experiment.

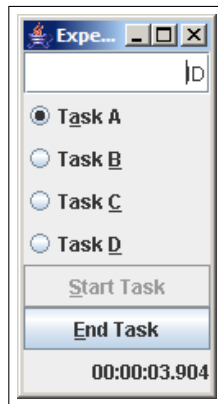


Fig. 3.7: **ExperimentMenu** handles study logging/recording duties.

When an experiment is started:

¹⁰ See §4.3 for a further discussion of the XML log schema used.

1. An XML log is opened, and populated with a log ID and task information
2. The ExperimentMenu window is minimised, due to a Java interface bug which caused the timer to show up even when the screen was in the background
3. The MemoryFrame window is given focus and set to full screen
4. The MemoryFrame window is populated with cards from a config file pertaining to the selected task
5. A timer is started, and the start time logged

When an experiment is stopped:

1. The timer is stopped, and time taken (H_1) is recorded
2. Per-card statistics (start/end positions, distance travelled, times clicked) are recorded
3. The XML log is finalised and closed

3.4.3 *Supporting Multi-Mouse Interaction*

Most desktop applications cannot support interaction from more than one mouse at a time. Standard behaviour when Windows is confronted with input from two or more USB mice is to treat them all as having an equal effect on the system cursor. The HCIL MID library mentioned earlier was developed for use in Win98/ME, and the techniques employed there could not be used in Windows XP due to changes in the input handling structure in the newer OS.

In Windows XP, such handling must be done through the `RawInput`¹¹ interface. Thankfully, Jake Stookey's C `RawMouse` library¹² provides a more developer-friendly API into this part of the operating system than Microsoft provides.

In order to use the raw mouse input, the `MouseClient` program developed for this study polls the operating system for available raw input devices, determine which of those are mouse devices, register to receive input events from them and then set up a callback to receive input events. By default an application will only receive events when it has user focus; registering with the `RID_INPUTSINK` flag allows the application to receive events in a global context, so that it can still capture mouse input when the `Java Memory` application is in the foreground.

Mouse movement is received in the form of relative $(\Delta x, \Delta y)$ events, so the application must hold state information for each mouse, consisting of a virtual cursor (automatically clipped at screen viewport boundaries) and button states.

An ICE network socket connection is set up between `Memory` and `MouseClient`. As mouse movements and click events are received by the callback function, `mouseMoved()`, `mouseClicked()`, `mouseDragged()` and `mouseReleased()` events are dispatched to the `MouseInputManager` class as appropriate.

¹¹ <http://msdn.microsoft.com/en-us/winui/winui/windowsuserinterface/userinput/rawinput.asp> [2006]

¹² <http://link.mywwwserver.com/~jstookey/arcade/rawmouse/> [2005]

4. INTERACTION DATA VISUALISATION AND ANALYSIS TOOL

The study of human action patterns is an important feature in the field of human–computer interaction. By collecting data within an information appliance (or collection of appliances) and replaying it along with human data collected through more conventional audio/visual recording, we may be able to provide a richer analysis of the human–computer system than each method could provide by itself [13]; allowing researchers to better contextualise and explain motivations for users’ actions, and identify specific areas in which they run into difficulties. It can also help to disambiguate comments made during a recording, providing additional supportive information in environments where video coverage is not adequate to explain what is happening (e.g. subtle actions, actions made that are occluded by users’ bodies, etc.).

4.1 *Problem Domain*

The data captured by the computer vision system is user–agnostic: it cannot automatically tell which of several users is touching the screen at any given point. As such, there needs to be an external way of mapping interactions to users for analysis purposes.

For this purpose the interaction studies were taped on video, and analysed

to classify those interactions. While it would be possible to conduct this video analysis and update the XML capture data manually, such a task would be intensely time-consuming and prone to human error.

XML data structures are useful for storage, but they are not generally a useful format for visualising interaction data. The ability to play back and visualise spatio-temporal data concurrently with video playback enables much faster classification of interactions by participant; it also may provide insights into the way that people perform interactions by allowing more natural “eye-balling” of logged data.

4.2 *Tool Requirements*

When constructing the tool, the following requirements were considered to be important in our classification task:

- The ability to visualise XML data logs temporally and spatially
- The ability to view/browse video streams
- Synchronising video streams and log data
- The ability to select interaction data structures and apply a participant ID appropriately
- The ability to save the changed XML into an appropriate file for further analysis

Additionally, the ability to split and/or merge interactions at specific points, to allow for post-hoc error correction in data, where appropriate was considered useful, but given a low priority due to time constraints.

Note that only the CV system interaction data was classified. Each mouse was assigned a unique ID when used, making this analysis unnecessary. The physical case was not included in this per-user analysis because the method does not require users to keep objects in contact with the screen where they could be tracked by the CV system — no meaningful data could be extracted from the single camera recording in the experimental space available.

4.3 XML Log Schema

Each log is given a unique alphanumeric identifier when recorded. Along with interaction data, information pertaining to the task is stored to aid later analysis. This includes the task ID (“**config1**” thru “**config4**”, pertaining to T_A – T_D respectively), method (“single”, “multi”, “mouse” or “physical”), and collections of card pairs grouped by their initial positioning on the screen (on left-hand side, right-hand side or split between the two).

```
<loginfo>
<log id="#alphanumeric">
<task id="#task" method="#method"/>
  <pairs type="LL">
    [multiple] <card id="#c"/>
  </pairs>
  <pairs type="LR">
    [multiple] <card id="#c"/>
  </pairs>
  <pairs type="RR">
    [multiple] <card id="#c"/>
```

```
</pairs>
```

```
</task>
```

The start and end times of the interaction are logged. Each individual **press**, **drag** and **release** entity contains a unique ID, a timestamp, positional information and the card associated with that interaction event. After classification (or in the mouse case), each will also contain one participant ID.

```
<start time="#hh:mm:ss.sss"/>
```

```
[multiple of]
```

```
[+]<press id="#id" time="#hh:mm:ss.sss" x="#x" y="#y">
```

```
    [?]<participant id="#p"/>
```

```
    <card id="#c"/>
```

```
</press>
```

```
[+]<drag id="#id" time="#hh:mm:ss.sss" x="#x" y="#y">
```

```
    [?]<participant id="#p"/>
```

```
    <card id="#c"/>
```

```
</drag>
```

```
[+]<release id="#id" time="#hh:mm:ss.sss" x="#x" y="#y">
```

```
    [?]<participant id="#p"/>
```

```
    <card id="#c"/>
```

```
</release>
```

```
[+]<match card="#c" with="#c" time="#hh:mm:ss.sss"/>
```

```
<end time="#hh:mm:ss.sss"/>
```

Finally, per-card statistics are gathered: the starting and ending positions on the canvas, the number of times each card was interacted with, and the total distance travelled during interactions. The log file is then closed.

Each `<log/>` can be concatenated after classification for analysis over multiple logs.

```
<cards>
  <card id="#c">
    <cardstart x="#x" y="#y"/>
    <cardend x="#x" y="#y"/>
    <cardclicked times="#t"/>
    <carddistance distance="#d"/>
  </card>
</cards>
</log>
</loginfo>
```

Note that by using a generalised schema for the events, we were able to log `devicePressed`, `deviceDragged` and `deviceReleased` events for either screen or mouse interactions regardless of their architectural differences [11]. It is possible that, in a hybridised setting where more than one device is available, two specialised visualisation methods could be provided on the same canvas using the same general schema as a basis, but enriched with device-specific information (e.g. hand bounding boxes) where appropriate.

4.4 Visualisation

The application provides two windows for the researcher: `VisFrame` is the main visualisation canvas, and `ControlPanel` provides the ability to browse through data and classify it.

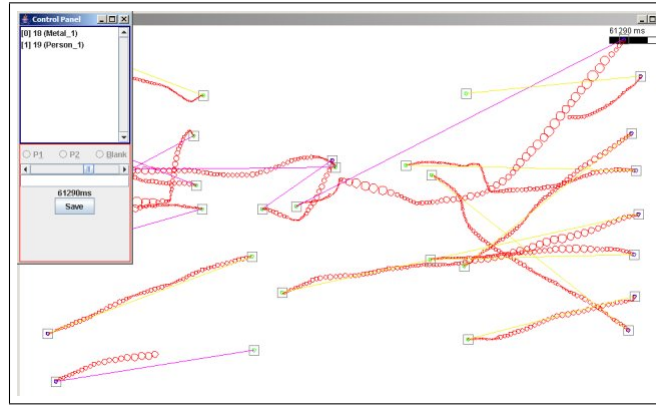


Fig. 4.1: Visualisation tool early in development.

Data is collated using unique interaction IDs prior to visualisation. The contents of the **VisFrame** canvas at an given point in time is determined primarily through a temporal cursor: only interactions that start before the point in time specified by this cursor will be displayed, whether in part or in full.

When the time cursor passes the start time of an interaction, the start and end points will be displayed (blue and green circles respectively). A line is drawn between the start and end points, giving the researcher an idea of where the interaction being started will end up. This line is coloured depending on the participant who performed this interaction. As the time cursor progresses, individual **drag** actions are be plotted on the canvas as red circles. The radius of the circles drawn is relative to the amount of distance travelled since the last **drag** action, giving the researcher some indication of the velocity¹. The radius of these circles is capped at a certain level (20 pixels), as the larger circles that may result from sudden positional disjunctions due to noisy data (or the single-touch interface averaging between two interactions) could significantly reduce

¹ This is not strictly indicative of the real-world velocity of an interaction, relying as it does on a fixed sampling interval between actions.

the clarity of the visualisation through clutter.

Interactions that are currently active (i.e. the time cursor lies between the start and end points of the interaction) are displayed in a list box the `ControlPanel` (see Fig. 4.2, where the researcher can select them. A selected interaction is further highlighted by drawing double squares around the start and end points, and drawing **drag** action circles in another colour (darker red) for contrast.

4.5 Clutter Reduction

Over a long period of logging, certain areas of the screen may become quite cluttered due to repeated movements spanning those areas. It may become progressively more difficult for the researcher to determine which **drag** circles belong to which interactions.

With this in mind, the time cursor is used to describe a Temporal Region of Interest (TRoI) within the data for the purpose of clutter reduction. As time passes and data begins to lose its temporal relevance, the colour of the **drag** circles fades towards the background colour.

The amount of fading relative to the time cursor position is specified by a combination of four constants. `FADE_STAY` determines the length of time that will pass after the end of an interaction before fading occurs (5s in the current prototype) – effectively the width of the TRoI. Once that period has passed, interactions will fade, blended between the draw colour and the background colour using level parameters `MIN_FADE(=0.0)` and `MAX_FADE(=0.9)`, over the period `FADE_OVER` (10s). Blending occurs in a linear fashion, governed by:

$$\begin{aligned}
\text{FadeColour}(r, g, b)_t &= \text{fade}_t. \text{DrawColour}(r, g, b) \\
&+ (1 - \text{fade}_t). \text{BackgroundColour}(r, g, b)
\end{aligned} \tag{4.1}$$

$$\text{fade}_t = \begin{cases} \text{MIN} & \text{if } \Delta t \leq \text{STAY} \\ \text{MAX} & \text{if } \Delta t \geq (\text{STAY} + \text{OVER}) \\ \text{MIN} + \frac{(\text{MAX} - \text{MIN}) \cdot (\Delta t - \text{STAY})}{\text{STAY} + \text{OVER}} & \text{otherwise} \end{cases} \tag{4.2}$$

$$\Delta t = t_{\text{Cursor}} - t_{\text{InteractionEnd}} \tag{4.3}$$

Note that selected interactions are not subject to fading, and are highlighted as usual. This allows a specific interaction to remain “fresh” if the researcher deems it relevant to data that occurs later in the visualisation.

It is possible that what is considered “temporally relevant” may change over the period of data analysis. Future implementations of this software may include sliders that allow these parameters to be dynamically changed, and possibly support non-linear fading. Another possibility is that different kinds of data may remain relevant for longer than others. In this case, fading may occur with different parameters governing each class of data.

4.6 Data Browsing and Classification

The classification tool supports three major methods of browsing data. Firstly, the time cursor may be manually adjusted using a `ControlPanel` scrollbar, or moved back and forth in small steps using key bindings (`'z'` and `'b'` respectively). The granularity of stepping is determined by a drop box, offering step sizes of

1, 5, 10, 50, or 100. Each unit of step size corresponds to an interval of 30ms, approximately the sample rate of the mouse device events.

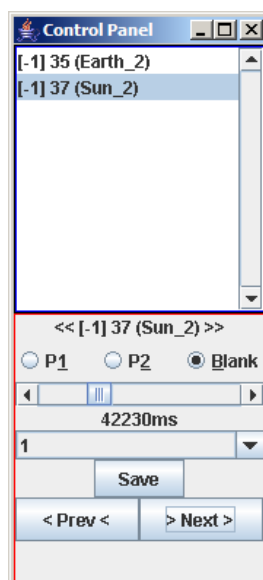


Fig. 4.2: ControlPanel in detail.

Secondly, real-time playback may be controlled by key bindings, allowing logs to be played back in parallel with video data. Play, Pause and Stop/Rewind functions are supported ('v', 'x', 'c' respectively)², giving functionality similar to the VCR-like [13] controls found on many video playing applications.

Thirdly, the [Next] and [Prev] buttons on the ControlPanel allow the researcher to quickly flick back and forth between interactions. When an interaction selected in this method, the time cursor travels to the end point of that interaction.

Classification of data by participant is achieved by selection a current interaction, then clicking either the [P1] or [P2] radio button. The [Blank] button

² Key bindings were chosen by their similarity with the music playing/browsing keys used in the WinAmp MP3 player: <http://www.winamp.com/> [2006]

is included for situations where an interaction has been found to be erroneous, such as a “false positive” error caused by an external IR light source. After all interactions have been appropriately classified, the research selects the **[Save]** button, and an XML data file is exported.

Unfortunately, one special case was not able to be properly classified by this tool. In the single-touch screen case, an interaction might be started by one participant and then either interfered with or continued by another, but the logging system would record it as only one interaction. These cases were handled manually, with the researcher editing the XML log files and splitting them into separate interactions: An interaction with an ID of “5” might be split into several segments, becoming interactions “5.1”, “5.2” and so on, each with a separate participant ID. Periods of interaction by two users at once required duplication of a sequence of **drag** elements, one for each contemporaneous participant.

Thankfully this situation did not occur very often over the course of the interaction study, except in the one case (E05, Task 4) where the users treated the “bug” of multiple touches being averaged on the single-touch screen as a “feature”, enabling them to quickly move cards around the screen. In this case, 11 such incidents happened within a 2-minute period.

4.7 Performance and Possible Improvements

The process of data classification for this study was greatly quickened by the application of this tool. In the worst case mentioned above, requiring the manual adjustment of 11 interactions, took one hour and ten minutes for a 2-minute task, or a classification time ratio of 35:1. In most ($\approx 95\%$) cases, classification

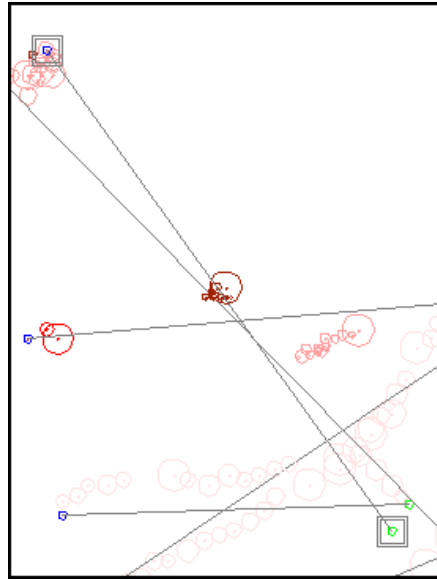


Fig. 4.3: A number of “worst case scenario” interactions on a single-touch screen log.

of participants in the screen methods yielded ratios of between approximately 1.25:1 and 2:1. Once classified, data to support hypotheses H_1 thru H_5 were extracted in tab-separated text format for 40 logs files in approximately nine seconds.

Some improvements can still be made to this classification process. Additional screen real estate would have helped; a second monitor would have allowed the researcher to view video data and interaction log data at the same time, rather than having to switch between applications. Alternately, an application which displayed video data next to a scaled-down representation of the screen canvas might prove useful (though again, more physical screen size/resolution would make the job easier, with less chance of losing small details).

Automation of our “worst case” events is possible. A researcher could set “mark-in” and “mark-out” events, much like those in professional video editing

software such as Adobe Premiere³, in the period where an interaction on a single-user screen is shared. Upon clicking a “Split” button, that interaction could be automatically segmented and given new sub-ID strings, which could then be classified just as normal interactions currently are.

As a visualisation tool, there are adjustments that could be beneficial for certain kinds of data, also. Using the tool, CV positional data was shown to be quite noisy, especially in areas where the screen sensitivity was lacking. An option allowing the smoothing of **drag** action positions, or visualising positions alongside a smoothed curve showing where they deviate from the general path, may be useful for certain kinds of analysis.

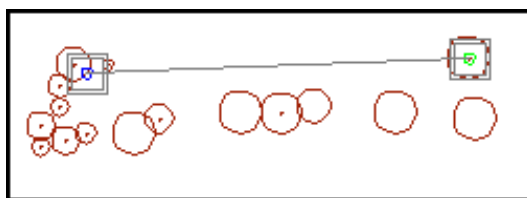


Fig. 4.4: Data from a noisy screen interaction.

Another possible option is moving away from circles and using ellipses or teardrop shapes, allowing the directional velocity of any given **drag** action to be visualised, rather than just a scalar value. This may allow a researcher to investigate flow patterns across a number of interactions or collections of user data for certain task domains.

If interaction velocity was a property that was of particular interest to the researcher, a non-linear relationship between velocity and circle radius could be employed. A logarithmic function could allow for more readable visualisation of fast or highly noisy data without the current “capping” technique, though the

³ <http://www.adobe.com/products/premiere/> [2006]

magnitude of smaller actions would be reduced by comparison.

5. RESULTS

This chapter presents the results of the interaction study, detailing analysis of data, hypothesis testing, questionnaire results and general observations.

Due to time, health and scheduling constraints, only ten of the proposed 16 experiments were completed ($E_{01}-E_{08}, E_{12}, E_{14}$). This precludes the originally intended balanced analysis of order, task and method factors. The lack of adequate incentives made to participants is considered to be a major contributing factor — many potential participants simply did not turn up when they were supposed to. However, sufficient data has been acquired to at least compare method means across the various data items captured.

5.1 *Demographics*

Of the 20 participants, 13 were male and 7 female. Ages ranged from 19 to 33, with an average of 26.15. 75% of participants were students (65% postgraduate, 10% undergraduate; 50% Science/Engineering and 25% Art/Design). One participant was left-handed, two described themselves as ambidextrous and the rest were right-handed.



Fig. 5.1: Good hand recognition

5.2 Commentary on Computer Vision System

Unfortunately, the computer vision system did not behave ideally during the study. Just prior to engaging in the study it was found that the projector had been moved and the LED panels had been knocked out of alignment. While the panels and projector were readjusted as much as possible over the course of the few hours left before to the first pair of participants were scheduled to arrive, it was decided to carry on with the CV calibration in a less than perfect state, due to the lack of time remaining for the study. As such, image registration on the screen cases was significantly less accurate than that available earlier in the course of research, with cursors being as much as 5cm out of alignment at the corners of the screen. No further positional adjustments were undertaken throughout the user study, as that would have created a confounding factor in data analysis.

Through most of the surface, registration of hand shapes was clear and

responsive (Fig. 5.1). However, in two areas in the bottom corners, uneven illumination caused limited responsiveness. Many of the cases where participants “lost” cards they were dragging occurred in these areas (Fig. 5.2).

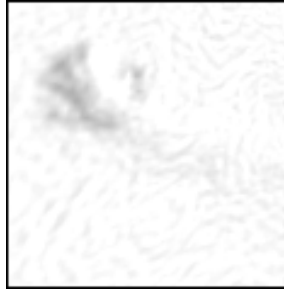


Fig. 5.2: Bad recognition due to uneven illumination

Another observation noted with the system is that, while picking up a card with the palm of the hand was usually done without problems, users often wished to place cards by “pushing” them using only the fingers of the hand. This led to marginal detection of hand position which exacerbated tracking problems in areas in which illumination problems occurred (Fig. 5.3).

5.3 Log Data Analysis

In this section, each of the five major data-based hypotheses are addressed and analysed.

5.3.1 Hypothesis H_1

There are no significant differences in the mean times taken to complete the task using different interface methods.

There was a strongly significant main effect for the amount of time taken T_x , with the multi-mouse (MM) case being clearly the fastest with a mean of 31.14

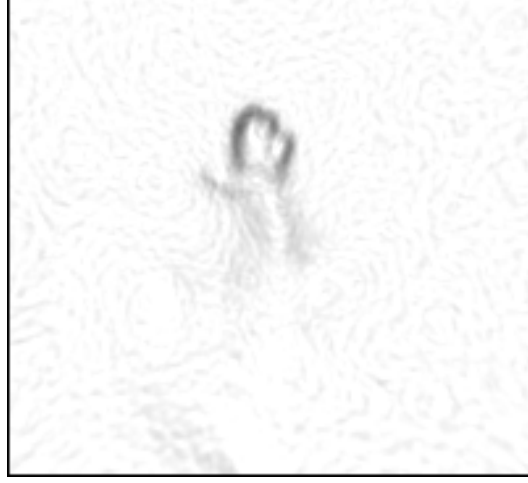


Fig. 5.3: Marginal recognition due to tilted hand posture

seconds ($\sigma = 7.35$). ($F_{3,36} = 58.315, p < 0.001$), allowing us to reject the null hypothesis H_1 . The difference of 12.25 seconds between the multi-touch (MT) and ST means was however not significant enough ($t_{18} = 0.921, p = 0.369$) to indicate that one method is clearly faster than the other.

While not balanced conditions, H_1 means for order and task were also compared. No significant learning effect ($F_{3,36} = 1.145, p = 0.344$) or task difference effect ($F_{3,36} = 0.623, p = 0.605$) were detected.

5.3.2 Hypothesis H_2

There are no significant differences in the mean times taken to complete the task using different interface methods.

There was no significant main effect in the mean proportions of time each participant spent using the interface ($F_{2,27} = 0.197, p = 0.822$) between the multi-mouse and screen methods.

However, given that the R_x value chooses the ratio of the minimally-participating

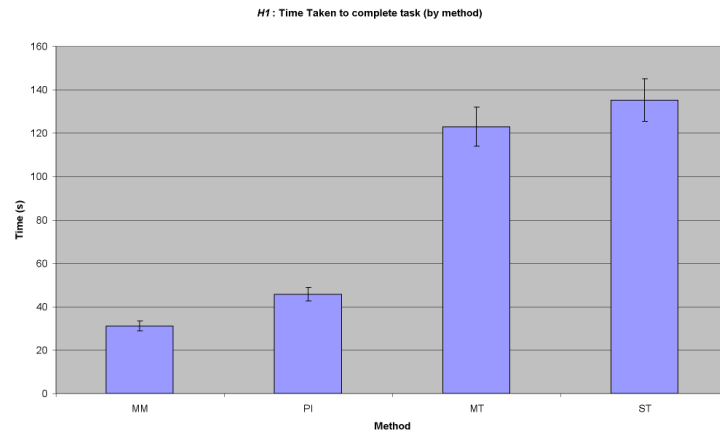


Fig. 5.4: H_1 : Time taken to complete task

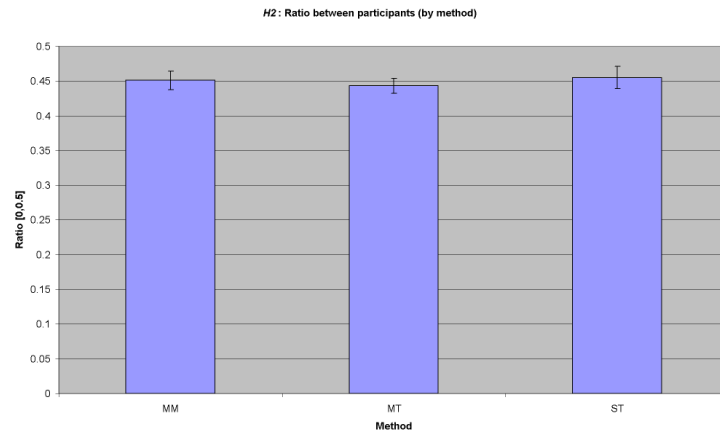


Fig. 5.5: H_2 : Ratio of participant interaction time

user, the value is not normally distributed. The test was revised, obtaining $R2_x$ values equivalent to the participation values of P_1 in each experiment over the full ratio continuum of $[0.0, 1.0]$ ¹. This test yielded a mean across methods of 0.502 ($\sigma = 0.066$). Again, no significant difference in means between methods was found ($F_{2,27} = 0.744, p = 0.485$). We can safely accept the null hypothesis.

An open style of interaction was maintained throughout the experiment, regardless of input method used. This is consistent with the findings of Ryall et al. [28], where “groups of two very often worked closely with one another throughout the entire trial” in a tabletop collaboration task, while larger group collaboration efforts were more likely led by a single user.

5.3.3 Hypothesis H_3

There are no significant differences in the average distance each card was moved using each interface method.

Using the ratio ($Sum(d1)/Sum(d2)$) as a measure of efficiency, there is a significant main effect in means between the three conditions ($F_{1,18} = 10.683, p < 0.005$), with the MM case being the most directed and the MT case being the least directed. We can reject the null hypothesis. The two screen cases were significantly different ($T_{18} = 3.337, p < 0.005$).

Upon reviewing video data, a possible explanation for this disparity appears to be that users working on the device simultaneously sometimes have to move around the actions of other users to avoid the CV system from confusing two close-by interactions for the same interaction, causing a user to “lose” a card. It worthwhile to note also that what may be “efficient” for a mouse-like device

¹ Doing the same with P_2 would produce a complementary distribution, so there is nothing to be gained by testing both.

may not necessarily be so for hand interaction, due to the ergonomics of human joint motion; moving the hand in a smooth horizontal line may take considerably more effort than pivoting at the shoulder or elbow and moving the hand in an arc.

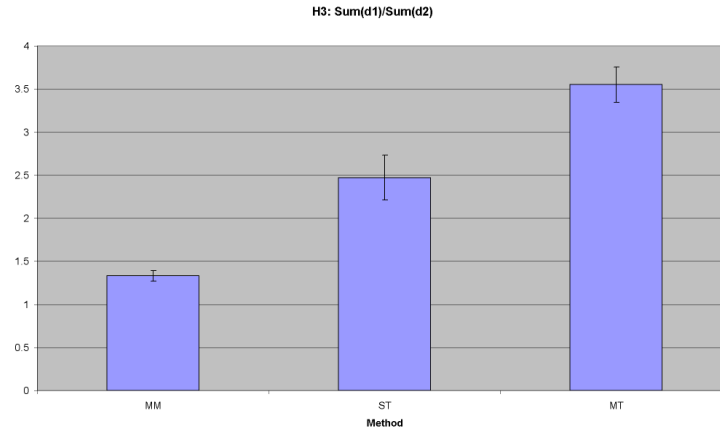


Fig. 5.6: H_3 : $Sum(d1)/Sum(d2)$ efficiency measure

5.3.4 Hypothesis H_4

There are no significant differences in the amount of time it takes to match pairs in (LL,LR,RR) configurations between each interface method.

A 3x3 ANOVA analysis of normalised matching time means showed no significant differences between methods ($F_{2,27} = 0.948, p = 0.4$) or between pairing location ($F_{2,54} = 1.303, p = 0.28$). This tends to indicate a balance of private and public tasks using the interface in each method.

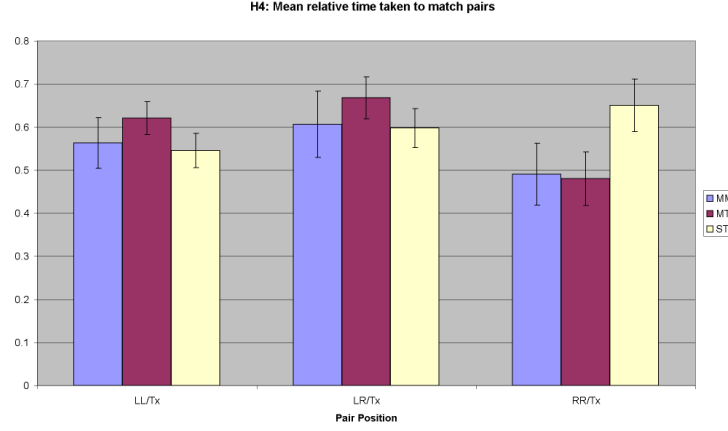


Fig. 5.7: H_4 : Normalised pair matching times

5.3.5 Hypothesis H_5

There are no significant differences in the amount of turn-taking that occurs between each interface method.

There was a significant difference found between methods in the amount of time spent in each state ($F_{3,116} = 183.941, p < 0.001$). To directly address the hypothesis, however, we need to look at the proportion of transitions of the categories mentioned in §3.3.2. Of the four categories mentioned, only the *Maintain* interaction does not constitute a form of turn-taking behaviour; each of *Handoff*, *Interrupt* and *Interject* are characterised by some degree of interplay between participants. Looking at the proportion of *Maintain* interactions compared to the other types, a significant difference was found ($F_{2,27} = 17.262, p < 0.001$).

In the ST case, 67.7% of interactions were maintained by a participant, compared to 36.0% in the MT case and 28.7% in the MM case. This indicates

that the highest level of turn-taking occurs in the MM case and the least in the ST case, with the MT case taking the middle ground. The multi-touch screen is certainly closer to modelling the turn-taking behaviour of the multi-mouse case, where users are free to act simultaneously with a minimum potential for disruption each other's actions.

Also note (Fig. 5.8) that in the ST case there are almost no Interrupt or Interject actions. This is to be expected, as the interface behaviour is not conducive to these actions being beneficial to the users, unless the “bug” of averaging positions is used as a “feature” to move a card from one interaction position to another (classified as an Interrupt action under this methodology, but performed as a collaborative operation.)

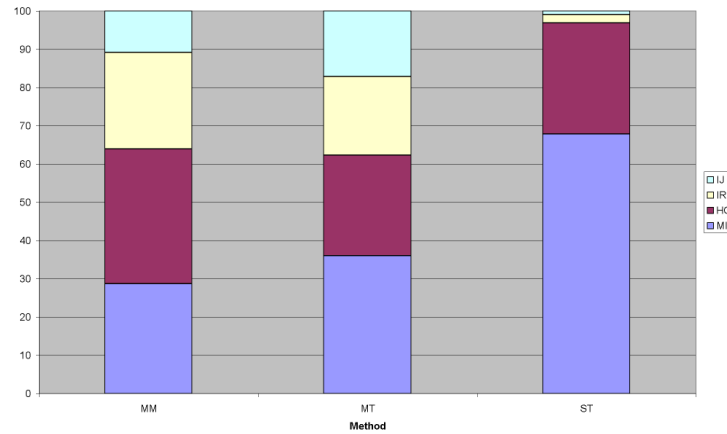


Fig. 5.8: H_5 : Comparison of turn-taking behaviour between input methods (MI: Maintain interaction, HO: Handoff, IR: Interrupt, IJ: Interject)

5.4 Questionnaire Data

Amongst the general biographical data, participants were asked how often the placed information on and referred to information in either computers or wall spaces. Fig. 5.9 shows that the majority of participants use computers far more often than they use wall spaces for their information needs. In the “short answer” section, two participants specifically indicated that the MM case was easier because they were familiar with the input method from daily use, even if the multi-user condition was novel.

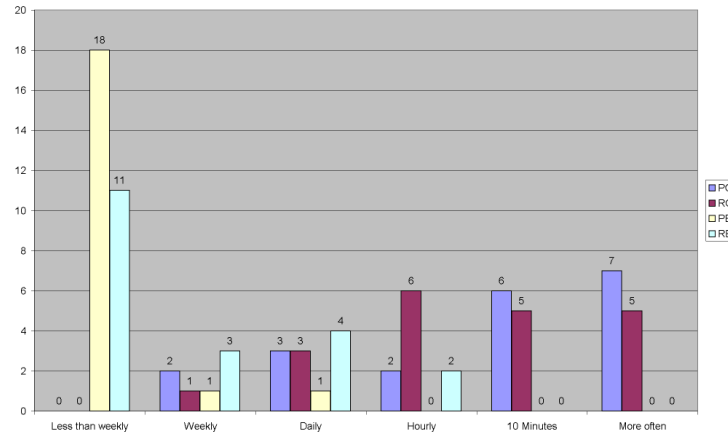


Fig. 5.9: Participant usage of wall spaces and computers
(P*: Put information on, R*: Refer to information; *B: Board, *C: Computer)

Of the questionnaire statements, 14 were found to contain significant differences in the level of agreement found between the four methods, while only 6 did not contain significant differences. These are described below, and each block of statements is summarised in Fig. 5.10, Fig. 5.11 and Fig. 5.12 respectively.

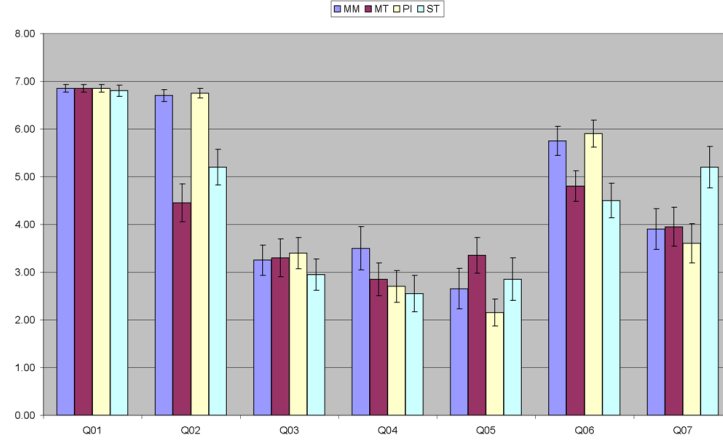


Fig. 5.10: Questionnaire Results Summary — (Q_{01} – Q_{07})

Q_{01} : *I found the task requirements easy to understand*

There was no significant difference between methods (Friedman test, $\chi_r^2 = 0.05, df = 3, p = 0.997$). Given the group median of 7, this indicates that the task was universally well understood across all cases.

Q_{02} : *I found the task easy to achieve*

There was a significant difference in ease of achievement (Friedman test, $\chi_r^2 = 27.20, df = 3, p < 0.001$). MM and PI cases were considered similarly very easy (medians of 7 each), with the screen cases being less easy (5 each). The ST case was considered marginally easier; looking at short answers, this appears to be due to more errors occurring during the MT case when two separate interactions were mistaken for each other by the CV system.

Q_{03} : *I did most of the work in performing the task*

There was no significant difference between methods (Friedman test, $\chi_r^2 =$

1.79, $df = 3, p = 0.618$). With a group median of 4, most participants felt neutral about the statement, indicating a feeling of work being shared relatively equally overall.

Q₀₄: I was distracted by what my partner was doing

Q₀₅: My input was disrupted by what my partner was doing

No significant differences were found between methods in either *Q₀₄* (Friedman test, $\chi_r^2 = 3.62, df = 3, p = 0.306$) or *Q₀₅* (Friedman test, $\chi_r^2 = 4.79, df = 3, p = 0.188$). With group means of 3 and 2 respectively, participants generally felt low levels of distraction from their partner's actions, or disruption of their own actions. Taken in concert with other factors suggesting a reasonable level of collaboration was taking place, this tends to indicate that, while there was some distraction and disruption, participants quickly learned to co-operate in ways that did not disturb each other.

Q₀₆: It was easy to use this interface with a partner

Q₀₇: It would be easier to use this interface by myself

There was a somewhat significant difference found between methods in *Q₀₆*. (Friedman test, $\chi_r^2 = 12.26, df = 3, p < 0.010$). PI and MM cases rated most highly (medians of 6) followed by MT (5) and ST (4.5). There was also a slightly significant difference in *Q₀₇* (Friedman test, $\chi_r^2 = 10.47, df = 3, p < 0.025$), with participants moderately agreeing that the ST case would be easier to use alone (median of 6) and neutral about the other cases (medians of 4 and 3.5).

These indicate that the ST interface may suffer somewhat in its ease of use from its lack of support for multiple interactions. The poorer performance of the MT interface relative to the PI and MM cases could be explained by the poor performance of the CV system in general.

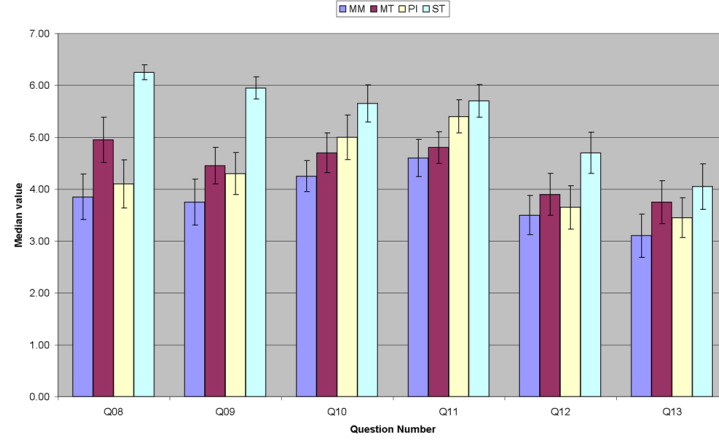


Fig. 5.11: Questionnaire Results Summary — (Q_{08} – Q_{13})

Q_{08} : *I feel that I paid attention to what my partner was doing*

Q_{09} : *I feel that my partner paid attention to me*

A significant difference was found between methods in both Q_{08} (Friedman test, $\chi_r^2 = 16.44, df = 3, p < 0.001$) and Q_{09} (Friedman test, $\chi_r^2 = 19.31, df = 3, p < 0.001$). Medians for each condition were identical across the two cases. The ST case rated the highest for attention (6) followed by MT (5), PI (4.5) and MM (4).

The high rating for the ST case suggests that participants take more care to watch what there partners are doing when interacting with the interface than in other cases, due to the interruptive consequences of two participants trying to act at once. On the other hand, in the MM case the participants are not bound by constraints of personal space and can easily reach and move cards on either side of the interface; they do not need to be as careful about what they

do with respect to their partners actions.

Q₁₀: I feel that I provided useful input to my partner

Q₁₁: I feel that my partner provided useful input

Some degree of significance was found in each of Q_{10} (Friedman test, $\chi_r^2 = 13.97, df = 3, p < 0.005$) and Q_{11} (Friedman test, $\chi_r^2 = 11.09, df = 3, p < 0.025$). In each case the ST case rated higher (6) than MM and MT (5). This may be an effect of turn-taking behaviour, and having to watch what your partner is doing while they are doing it. The PI case rated higher for Q_{10} (6) than for Q_9 (5). This may be due it being easier to see physical cards being moved with peripheral attention. Overall, there was agreement that both partners in the paired experiments were providing useful input.

Q₁₂: I feel that I could learn from observing my partner's actions

Q₁₃: I feel that I could demonstrate things to my partner

Neither Q_{12} (Friedman test, $\chi_r^2 = 6.41, df = 3, p = 0.093$) nor Q_{13} (Friedman test, $\chi_r^2 = 4.61, df = 3, p = 0.203$) yielded significant differences in rating. Group medians of 4 indicate that participants were neutral about whether they could demonstrate or learn using all interface methods. To be fair, this task was not geared towards either demonstrating to or learning from a partner, so it is unsurprising that these measures received a neutral response.

Q₁₄: The application was stable

Q₁₅: The application behaved predictably

Significant differences were found in both Q_{14} (Friedman test, $\chi_r^2 = 14.91, df = 3, p < 0.005$) and Q_{15} (Friedman test, $\chi_r^2 = 35.30, df = 3, p < 0.001$). The PI and MM cases both received high ratings for stability and predictability (7 each). The ST case was less stable (6) and predictable (5), and the MT case the least

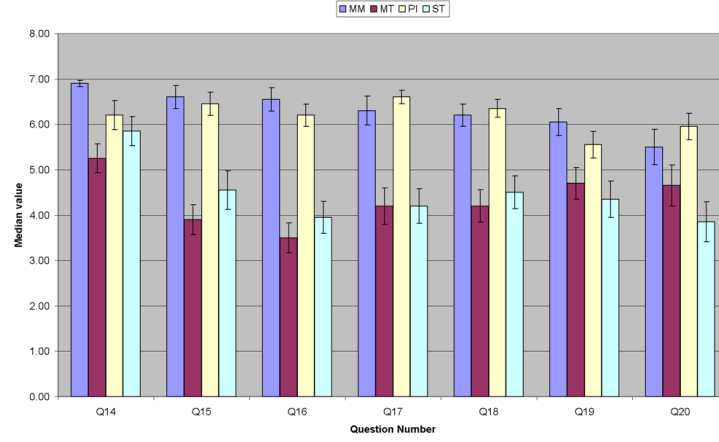


Fig. 5.12: Questionnaire Results Summary — (Q_{14} – Q_{20})

stable (5.5) and predictable (3.0).

While both screen cases were more difficult to predict than the other input methods, the MT case was seen as more unpredictable due to the CV system confusing two users’ interactions, or two interactions by the same user. When the tracking of two interactions merged, cards would sometimes “fly off” on odd directions; this was less likely to be experienced using the ST case which didn’t allow two simultaneous interactions. Participants feel they ought to be able to do more in the multi-touch case, but the limitations of the CV system become more apparent when they do try.

Q₁₆: It was easy to move cards around the task space

The ease of the dragging task was significantly different across methods (Friedman test, $\chi_r^2 = 41.12, df = 3, p < 0.001$), with MM and PI cases being the easiest (7 and 6.5 respectively), and MT and ST cases being more difficult

(3 and 4 respectively). Looking at short answers, the slight drop in the PI case appears to be due to the adhesive backing being slightly weak and cards dropping when a participant attempted to stick them to the screen.

Q₁₇: It was easy to correct any mistakes that were made

Participants generally (Friedman test, $\chi_r^2 = 34.94, df = 3, p < 0.001$) found it somewhat easy to correct mistakes in screen cases (medians of 5), though the MM and PI cases were significantly higher (7).

Q₁₈: It was easy to recover from any errors in the interface

Relatively similar results here (Friedman test, $\chi_r^2 = 23.09, df = 3, p < 0.001$). Strong agreement in the MM and PI cases (7), and slight agreement in MT and ST (4.5 and 5 respectively). While errors were still somewhat easy to recover from, the ability to do so still lags significantly behind the non-screen cases.

Q₁₉: This method provides a good way to arrange information

These results mirror the H_1 and Q_{02} results²: the significant differences (Friedman test, $\chi_r^2 = 14.15, df = 3, p < 0.005$) in Q_{19} show that MM, the quickest and easiest case to achieve, was also considered to best way to arrange information (7). Physical interaction showed a moderate degree of agreement (6), with the screen cases showing only slight agreement (5).

Q₂₀: This method felt like a natural way to solve the task

A significant difference between methods here as well (Friedman test, $\chi_r^2 = 13.43, df = 3, p < 0.005$). Unsurprisingly the physical interaction case rated the highest for natural interaction, followed by the familiar mouse case. The multi-touch screen received slight agreement with the statement (5), while the single-touch screen case received slight disagreement.

² A direct correlation was found between results in Q_{02} and Q_{19} (Friedman test, $\chi_r^2 = 7.2, df = 1, p < 0.01$), although at a lower confidence level.

A multi-touch screen environment certainly *feels* like a more natural way to interact, but the reliability of the CV system needs to be high enough to support what users feel like they ought to be able to do with it.

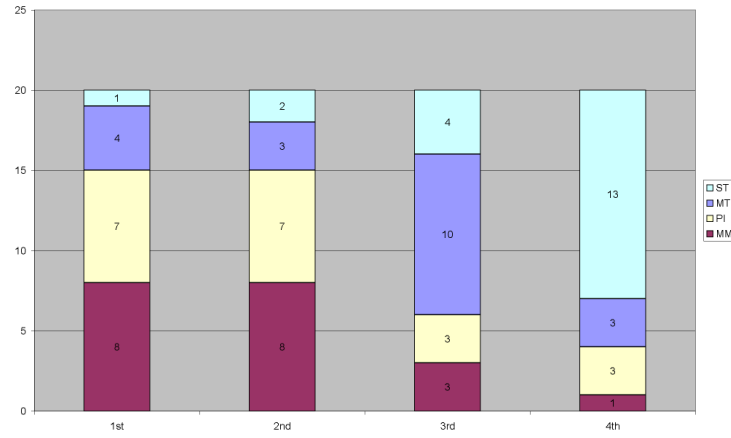


Fig. 5.13: Preference ranking of four methods

Uniquely ranking methods by preference yielded a significant result (Friedman test, $\chi_r^2 = 17.94$, $df = 3$, $p < 0.001$), shown in Tab. 5.1 and Fig. 5.13). The multi-mouse and physical conditions were nearly tied for first place, with the mouse method edging slightly ahead. The multi-touch condition was a clear third place, with 50% of participants ranking it there at third versus 35% ranking it first or second. The single-touch condition was the least preferred, with 65% of participants rating it last and only 15% ranking it first or second..

Tab. 5.1: Ranking interface methods by preference

Rank	MM	PI	MT	ST
1st	8	7	4	1
2nd	8	7	3	2
3rd	3	3	10	4
4th	1	3	3	13
N	20	20	20	20
Mean	1.85	2.1	2.6	3.45
Median	2	2	3	4
S.D.	0.875	1.07	0.995	0.887

5.4.1 Observations

A number of observations were gleaned from the “short answer” sections of questionnaires and video footage. The following is a summary of points raised from responses, and ranked questionnaire statements they pertain to where appropriate:

Screen Conditions (MT,ST)

- Image registration was a definite issue. Better calibration is needed
- Cards sometimes “jump across” the interface when one interaction starts just as another stops. The tracking algorithm needs to cull on distance more aggressively to avoid this
- When releasing cards, users still have a tendency to “drop” their hand from the screen, even after training
- Reliability affects attention. Users are too busy paying attention to ensuring that the cards go where they want them to $[Q_{02}, Q_{08}, Q_{09}]$
- A number of participants suggested some form of delay and/or visual cue be added to the ST system before the averaging of two users’ interaction

positions. While this would be difficult to implement directly on an ST surface, it might prove an interesting UI policy to explore when using an MT display in a single-touch or “shared interaction” mode

Multi-Touch Screen (MT)

- Disruption occurs when hands move too close together [Q_{05}]
- As well as two users being able to interact at once, being able to interact bi-manually is also useful
- “Fun, but difficult.” (S_{12})
- “Still in its infancy.” (S_{28})
- MT capabilities are missed when participants move to ST case

Single-Touch Screen (ST)

- It was sometimes difficult to see the cursor when it was under a hand
- Having to work with a partner makes ST more difficult [Q_{07}]
- ST feels unnatural: “I felt handicapped by the one-at-a-time way of working.” (S_{11})
- Experiences less disruption than MT due to more explicitly turn-tasking behaviour. [$Q_{05}, Q_{08} \dots Q_{13}$]

Multi-Mouse (MM)

- Fast cursor and card movements are disorienting and difficult to track on a large screen [$Q_{12}Q_{13}$]
- Participants found the familiarity of a mouse beneficial

- “I could steal cards easier.” (S_{03})

Physical Interaction (PI)

- Better adhesive needed for sticking cards to screen [Q_{14}]
- The ability to hold a card in your hand is beneficial when looking for a matching card

Physical Issues

- Having the whole screen wobble when you pushed hard was disturbing. A more rigid frame for the screen is desirable
- When sitting close to the screen, it is hard to get a good overview of what is happening on the screen
- Participant height affects usable screen area. Short participants had difficulties with reaching higher cards, or reaching across past the table used for the MM case. Further study should take care to either use taller participants or a lower screen, and possibly have the mouse table a greater distance from the screen
- Ergonomic technology design needs to allow for a range of body types:
“There are no short, large-breasted women in the future?” (S_{23})
- Using the palm flat against a vertical wall is not particularly natural, but it might be more so if the screen were tilted backward
- Repeated, concentrated actions are tiring (including a lot of mouse clicking)

Task Comments

-
- A number of participants felt that audio feedback might have been beneficial in computer-supported tasks
 - A possible character/language bias exists in using *kanji*
 - The sense of ownership of the sides of the interface was high, just from the way the physical test environment was set up
 - Less side-ownership occurred in the MM case
 - Some users found mouse “stealing” disruptive and/or distracting
 - Generally easily understood
 - Generally considered a fun task

Even while not performing at its most reliable, the multi-touch screen shows some promise as an interaction method. It provides some of the natural engagement and direct manipulation affordances of physical interaction while at the same time allowing users to interact simultaneously, unlike the single-touch screen. The screen lends itself well to a two-person group using it in an open style.

However, there still is definite room for improvement. In some cases two people (or two hand interactions) were confused by the system, and users lost hold of what they were interacting with as a result — thus discouraging users greatly from using the device in a multi-touch fashion. Improvements need to be made to improve reliability, to make the system more consistent and less vulnerable to the foibles of calibration and illumination issues.

6. CONCLUSIONS AND FUTURE WORK

This research demonstrated the advantages and challenges of implementing a large-scale multi-touch interface. Two studies have been conducted: an in-depth pilot study in the lab and a long-term public trial, supported by a number of software tools developed over the course of this research. The computer vision system developed provides responsive, real-time multi-touch interaction, allowing many users to collaborate in face-to-face design tasks on the same device.

The user studies undertaken showed that providing an engaging, natural method by which users can interact simultaneously is beneficial compared to existing single-touch screen technologies; users found it less frustrating when they did not have to engage in explicit turn-taking. It also provides advantages over a multi-mouse condition in that users are less distracted by each other's interactions, finding them easier to follow with peripheral awareness. It also draws on the tangible affordances of physical interaction, while still supporting the migration of moving design artifacts into other devices and problem domains once the “big board” work is completed.

The user classification tool was found to significantly reduce the amount of time that the researcher spent in the video analysis phase of the interaction study. Further automation of the “difficult” cases is possible, and desirable if further study is to be performed using the software framework developed.

In creating the multi-touch screen technology, Jun Rekimoto's conclusion [24] that providing sensitivity to lighting conditions was a major hurdle in this kind of endeavour was confirmed. While the system could have benefited from more rigid and stable hardware construction, a better approach may be to swap out the current illumination and display surface in favour of technology such as Jefferson Han's FTIR technology [10]. Some research into screen materials and scalability may be required, but it should in theory be possible to retrofit this sensing technology to the existing prototype without significant changes in the CV system.

With higher levels of processor power becoming more available, it may be feasible to substitute a more complex tracking system such as Condensation filtering [15]. However, improvements can still be made within the lightweight approach currently being used. More intelligent predictive measures could come into place in situations where collision between two interactions is likely, so that extra processing is performed only when needed. In cases where two interactions do touch, tracking information and more in-depth contour detection techniques (such as exploiting "hole" regions within a contour, and drawing on a normalised source image for component connection) could allow for intelligent "splitting" of shapes, allowing the two interactions to preserve their individual identities.

Positional tracking could be greatly improved by increasing camera resolution. Use of a more expensive camera would allow frame capture at 640x480 resolution at the very least. In parts of the system where more complex functions are performed and the positional accuracy is less important, the image or image segments could be downsampled to a lower level of detail (e.g. 320x240) to improve efficiency. This would also allow for more reliable tracking of fingertip-

sized interactions and hand pose classification.

Many users were dissatisfied with the sensitivity of the screen being variable due to uneven illumination. Using a stereo camera setup [43] would provide more depth information to the touch interaction, increasing the consistency of touch sensitivity across the screen surface.

Image registration is also a major concern. A good approach to improving registration would be the implementation of an automatic calibration tool, such that an accurate viewport mapping could be constructed by touching the boundaries of the projected display space on-screen. While less important in a robust and self-contained hardware unit, it would make things a lot easier in the case where components of the system may be reappropriated for other uses at certain times.

Currently, the screen device is user agnostic; that is, it cannot determine which user belongs to which interaction. Through the use of other environmental sensors (e.g. overhead cameras) it may be possible to determine which user is performing which action. This would improve support for combining shared and private data on the same interface.

In theory, this sensing technology could be used in other orientations. Table-based interaction, or implementation as a tilted “draughting table” may also provide usable multi-touch surfaces. A possible avenue of future research is to determine what differences need to be made in the computer vision system when detecting and classifying interactions in different surface orientations.

In terms of the interaction study, it was found that the multi-touch screen has some way to go if it is to become as effective as existing mature input techniques – the technology felt to be “still in its infancy” according to some

participants. On the other hand, the tangible affordances and multi-touch experience do increase the level of engagement and “fun factor” involved.

Further research into what constitutes collaborative turn-taking behaviour on this kind of interface is desirable. However, in a system where we cannot rely on periods of multiple interaction being brief [29], a more in-depth analysis of what constitutes turn-taking behaviour and what is merely parallel interaction is required. The state model described in hypothesis H_5 can also to be extended to account for more than two simultaneous participants.

The task addressed in our pilot study was a short, simple one, where assumptions were made about the proportion of public to private tasks. It may be worth looking at a lengthier task, where the cognitive load of interaction is given less weighting than that of completing the task itself. A NASA TLX (Task Load Index) may be an appropriate measure to use in such a study; it would also give us some degree of differentiation between the physical and cognitive challenges of interaction that could inform the development of more ergonomic interaction techniques.

It may be worth looking at the inclusion of audio feedback when developing applications that use a large-scale screen. Positional information from the CV system could potentially be used to inform a stereo or 3D sound system within the environment, so that feedback sounds are situated relative to the interactions that caused them.

Other factors that could improve the classification process include incorporating the video stream into the same application rather than running the two side-by-side, allowing exact synchronisation between log data and the video feed. To accommodate different screen resolutions and form factors, the visual-

isation canvas could be rescaled to an appropriate size, so that the researcher does not need to switch between applications when two monitors or a larger screen is not available.

In terms of the visualisation itself, a number of possible improvements have been proposed. These include visualisations of data specific to certain input devices (bounding boxes for CV interactions, pen angle and pressure for digital tablets, etc.), smoothing of noisy data for better identification of interaction paths, and moving from circles to ellipses or teardrops so that each event drawn on the canvas provides some indication of motion information.

For clutter reduction during visualisation, a fixed window is currently used for persistence and fading of old interactions. Providing the ability for the user to adjust the parameters of this window in the interface could help certain kinds of analysis. It may also be possible for an intelligent automatic adjustment of the window based on the frequency and duration of incoming events.

7. BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Ravin Balakrishnan and Ken Hinckley. The role of kinesthetic reference frames in two-handed input performance. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 171–178, New York, NY, USA, 1999. ACM Press.
- [2] Ravin Balakrishnan and I. Scott MacKenzie. Performance differences in the fingers, wrist, and forearm in computer input control. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 303–310, New York, NY, USA, 1997. ACM Press.
- [3] Thomas Baudel and Michel Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.
- [4] Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [5] Paul M. Fitts. The information capacity of human motor system in controlling the amplitude of movement. *APA Journal of Experimental Psychology: General*, 47(6):381–391, June 1954.
- [6] Brian R. Gaines. Modeling and forecasting the information sciences. *Information Sciences*, 3(22):57–58, 1991.

-
- [7] Brian R. Gaines and Mildred L. Shaw. A learning model for forecasting the future of information technology. *Future Comput. Syst.*, 1(1):31–69, 1986.
 - [8] Jörg Geißler. Shuffle, throw or take it! working efficiently with an interactive wall. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 265–266. ACM Press, 1998.
 - [9] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA, 2001. ACM Press.
 - [10] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM Press.
 - [11] Gregory S. Hartman and Len Bass. Logging Events Crossing Architectural Boundaries. In *IFIP TC13 International Conference, Human-Computer Interaction — INTERACT 2005*, Berlin/Heidelberg, September 2005. Springer Verlag.
 - [12] Karen Holtzblatt. If we're a team why don't we act like one? *interactions*, 1(3):17–20, 1994.
 - [13] Lucio Ieronutti, Roberto Ranon, and Luca Chittaro. High-level visualization of users' navigation in virtual environments. In *IFIP TC13 International Conference, Human-Computer Interaction — INTERACT 2005*, Berlin/Heidelberg, September 2005. Springer Verlag.

-
- [14] Kori Inkpen, Joanna McGrenere, Kellogg S. Booth, and Maria Klawe. The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments. In *Proceedings of the conference on Graphics interface '97*, pages 138–145. Canadian Information Processing Society, 1997.
 - [15] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
 - [16] Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Mark Bilezikjian, and James A. Landay. The designers' outpost: a tangible interface for collaborative web site design. In *UIST*, pages 1–10, 2001.
 - [17] Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Raecine Meza, and James A. Landay. A tangible evolution: System architecture and participatory design studies of the designers outpost. CS Technical Report UCB//CSD-00-1117, UC Berkeley Computer Science Division, 2000.
 - [18] Nobuyuki Matsushita and Jun Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 209–210. ACM Press, 1997.
 - [19] Thomas P. Moran, Eric Saund, William Van Melle, Anuj U. Gujar, Kenneth P. Fishkin, and Beverly L. Harrison. Design and technology for collaboration: collaborative collages of information on physical walls. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 197–206, New York, NY, USA, 1999. ACM Press.

-
- [20] Meredith Ringel Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Beyond "social protocols": multi-user coordination policies for co-located groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 262–265, New York, NY, USA, 2004. ACM Press.
 - [21] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: new dimensions in office whiteboards. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 346–353. ACM Press, 1999.
 - [22] Mark W. Newman, James Lin, Jason I. Hong, and James A. Landay. DENIM: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction*, 18(3):259–324, 2003.
 - [23] Jun Rekimoto. Pick-and-Drop: a direct manipulation technique for multiple computer environments. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39. ACM Press, 1997.
 - [24] Jun Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120. ACM Press, 2002.
 - [25] Jun Rekimoto and Nobuyuki Matsushita. Perceptual surfaces: Towards a human and object sensitive interactive display. In *Workshop on Perceptual User Interfaces (PUI '97)*, 1997.
 - [26] Meredith Ringel, Henry Berg, Yuhui Jin, and Terry Winograd. Barehands:

-
- Implement-free interaction with a wall-mounted display. CS Technical Report TR-2000-04, Stanford University, 2000.
- [27] D.M. Russell, J. Trimble, and R. Wales. Two paths from the same place: Task driven and human centered evolution of a group information surface. In *Make IT Easy Conference*, 2002.
- [28] Kathy Ryall, Clifton Forlines, Chia Shen, and Meredith Ringel Morris. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 284–293, New York, NY, USA, 2004. ACM Press.
- [29] H. Sacks, E. A. Schegloff, and G. Jefferson. Systematics for the organization of turn-taking for conversation. *Language: Journal of the Linguistic Society of America*, 50(4):696–735, 1974.
- [30] Thomas Seifried. Future Office: A networked mixed reality framework. Bakkalaureatsarbeit, FH-Hagenberg Medientechnik und Design, January 2005.
- [31] Garth B. D. Shoemaker and Kori M. Inkpen. Single display privacyware: augmenting public displays with private information. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 522–529, New York, NY, USA, 2001. ACM Press.
- [32] M. Stark and M. Kohler. P.G. ZYKLOP: Video based gesture recognition for human computer interaction. In W. D. Fellner, editor, *Modeling - Virtual Worlds - Distributed Graphics*. Infix Verlag, 1995.

-
- [33] Jason Stewart, Benjamin B. Bederson, and Allison Druin. Single Display Groupware: a model for co-present collaboration. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 286–293. ACM Press, 1999.
- [34] Jason Stewart, Elaine M. Raybourn, Ben Bederson, and Allison Druin. When two hands are better than one: enhancing collaboration using single display groupware. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 287–288, New York, NY, USA, 1998. ACM Press.
- [35] G.M. Stratton. Vision without inversion of the retinal image. *Psychological Review*, 4:341–360,463–481, 1897.
- [36] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-LAND: an interactive landscape for creativity and innovation. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 120–127. ACM Press, 1999.
- [37] S. Suzuki and K. Abe. Topological structural analysis of digital images by border following. *CVGIP*, 30(1):32–46, 1985.
- [38] Peter Tandler. The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments. *Journal of Systems and Software*, (special issue on Ubiquitous Computing edited by J.J. Barton, R. Cerqueira and M. Fontoura), 22(3):267–296, 2003.
- [39] Peter Tandler, Norbert Streiz, and Thorsten Prante. Roomware — moving towards ubiquitous computers. *IEEE Micro*, 22(6), 2002.

-
- [40] Edward Tse and Saul Greenberg. Rapidly prototyping single display groupware through the sdgtoolkit. In *CRPIT '04: Proceedings of the fifth conference on Australasian user interface*, pages 101–110, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
 - [41] Alessandro Valli. White paper: The design of natural interaction, 2006.
 - [42] Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Effects of display position and control space orientation on user preference and performance. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 309–318, New York, NY, USA, 2006. ACM Press.
 - [43] Andrew D. Wilson. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, New York, NY, USA, 2004. ACM Press.
 - [44] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2003. ACM Press.
 - [45] Andrea Zanella and Saul Greenberg. Reducing interference in single display groupware through transparency. In *5th European Conference on Computer-Supported Cooperative Work (ECSCW'01)*, 2001.
 - [46] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision (ICCV '99)*, pages 666–673, 1999.

-
- [47] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):1330–1334, 2000.

APPENDIX

A. QUESTIONNAIRE FORMS

This appendix contains the forms used in the interaction study:

1. Information Sheet
2. Consent Form
3. Questionnaire Rubric
4. Pre-Experiment Questionnaire
5. Post-Task Questionnaire
6. Post-Experiment Questionnaire

University of Canterbury
Department of Computer Science and Software Engineering (CSSE)

Information

You are invited to participate as a subject in the research project “*Observation and Comparison of Interaction Methods using Large-Scale Interactive Displays.*”

The aim of this project is to determine how well a large, multi-touch interactive screen models the ways in which we use traditional large wall display spaces like whiteboards and bulletin boards, and observe the ways in which people collaborate using this kind of space. You and a partner will be asked to perform four simple pair-matching tasks (matching the symbols on pairs of cards) with a variety of interfaces. You will be given a short questionnaire to fill out at the end of each task, and there will be a short informal interview after all tasks have been completed.

You will be videotaped throughout this process. With your express permission, the researcher may keep excerpts or transcripts from videotapes for publication or for use in further research. Care will be taken to ensure your anonymity, and you will be given the opportunity to verify excerpts or transcripts prior to publication.

You have the right to withdraw from the project at any time, including withdrawal of any information provided. However, be aware that any information that has been contributed anonymously cannot be withdrawn after it has been collected by the researcher.

(turn over)

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: the identity of the participants will not be made public without their consent. To ensure anonymity and confidentiality, information will be stored in securely, only accessible to the researcher. Any transcripts will substitute the name of participants for fictional names, and any video excerpts used may use blurring or masking to hide the identities of the participants, if desired.

There are no risks to your physical or emotional health foreseen in participating in this project. If you feel that you are being put in any risk, or that there is any risk of cultural offense, please advise the researcher so that your concerns may be addressed.

The project is being carried out as a requirement for an MSc in Computer Science by David Thompson (who can be contacted at *[elided]*), under the supervision of Dr. Richard Green (who can be contacted at *[elided]*). He will be pleased to discuss any concerns you may have about participation in the project.

The project has been reviewed **and approved** by the University of Canterbury Human Ethics Committee.

Please take this information sheet with you when you leave.

*CONSENT FORM***Observation and Comparison of Interaction Methods using
Large-Scale Interactive Displays**

I have read and understood the description of the above-named project. On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved.

I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided.

NAME (Please print):

Signature:

Date:

Contact details (phone/email/etc.):

*Observation and Comparison of Interaction Methods using Large-Scale
Interactive Displays*

Please read the following note before completing the questionnaire.

The questionnaire is anonymous, and you will not be identified as a participant without your consent.

The questionnaire will be issued to you on several sheets over the course of the study process. Individual sheets will be collected together, but will not be added to other questionnaires until the end of the process

You may withdraw your participation, including withdrawal of any information you have provided, until your questionnaire has been added to the others collected. Because it is anonymous, it cannot be retrieved after that.

By completing this questionnaire, it will be understood that you have consented to participate in the project, and that you consent to publication of the results of the project with the understanding that anonymity will be preserved.

ID: _____

Biographical Information

Gender:

Age (years):

Occupation:

Course of Study (if student):

Handedness (Left/Right/Ambidextrous):

Height:

(turn over)

When you work/study, how often do you:

... put information into a computer? (tick one)

- Less than once a week* ☐
Once or twice a week ☐
Every day ☐
Every hour ☐
Every ten minutes ☐
More often than every ten minutes ☐

... refer to information on a computer? (tick one)

- Less than once a week* ☐
Once or twice a week ☐
Every day ☐
Every hour ☐
Every ten minutes ☐
More often than every ten minutes ☐

... put information on a whiteboard, noticeboard or similar public space?
(tick one)

- Less than once a week* ☐
Once or twice a week ☐
Every day ☐
Every hour ☐
Every ten minutes ☐
More often than every ten minutes ☐

... refer to information on a whiteboard, noticeboard or similar public space?
(tick one)

- Less than once a week* ☐
Once or twice a week ☐
Every day ☐
Every hour ☐
Every ten minutes ☐
More often than every ten minutes ☐

ID: _____

Task Assessment

Instructions: Please rate how strongly you agree or disagree with each of the following statements by placing a check mark in the appropriate circle. Use the following key:

Agreement	N°
Strongly Disagree	1
Moderately Disagree	2
Somewhat Disagree	3
Undecided	4
Somewhat Agree	5
Moderately Agree	6
Strongly Agree	7

Statement (1: strongly disagree ... 7: strongly agree)	1	2	3	4	5	6	7
I found the task requirements easy to understand	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I found the task easy to achieve	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I did most of the work in performing the task	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I was distracted by what my partner was doing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
My input was disrupted by what my partner was doing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to use this interface with a partner	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It would be easier to use this interface by myself	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that I paid attention to what my partner was doing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that my partner paid attention to me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that I provided useful input to my partner	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that my partner provided useful input	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that I could learn from observing my partner's actions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I feel that I could demonstrate things to my partner	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The application was stable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The application behaved predictably	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to move cards around the task space	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to correct any mistakes that were made	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to recover from any errors in the interface	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This method provides a good way to arrange information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
This method felt like a natural way to solve the task	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(turn over)

Please give a short answer to each of the following questions:

Was there anything in particular you felt was difficult about using this method to solve the task?

How could this interaction method be improved for you?

Any further comments?

ID: _____

Condition Ranking

Please rank the four interaction conditions used from 1 to 4, with '1' being your favourite condition and '4' being your least favourite condition. Use each number exactly once.

Condition *N*^o1 (Single/Multi/Physical/Mouse): []
Condition *N*^o2 (Single/Multi/Physical/Mouse): []
Condition *N*^o3 (Single/Multi/Physical/Mouse): []
Condition *N*^o4 (Single/Multi/Physical/Mouse): []

Any further comments?

B. STUDY RESULT TABLES

This appendix contains result tables from each of the hypotheses and ranked questionnaire statements.

Tab. B.1: H_1 : T_x means grouped by method.

Method	N	Mean	SD	Min	Max
MM	10	31.14	7.35	24.81	49.85
MT	10	122.96	28.55	65.61	154.19
PI	10	45.75	9.77	36.66	68.45
ST	10	135.21	30.91	84.93	177.31
Total	40	83.77	50.95	24.81	177.31

Tab. B.2: H_2 : R_x means grouped by method.

Method	N	Mean	SD	Min	Max
MM	10	0.451	0.042	0.347	0.492
MT	10	0.443	0.034	0.380	0.493
ST	10	0.455	0.051	0.367	0.499
Total	30	0.450	0.042	0.347	0.499

Tab. B.3: H_3 : $Sum(d1)/Sum(d2)$ means grouped by method.

Method	N	Mean	SD	Min	Max
MM	10	1.325	0.166	1.106	1.560
MT	10	3.552	0.636	2.813	4.977
ST	10	2.466	0.810	1.710	4.175
Total	30	2.448	1.092	1.106	4.977

Tab. B.4: Q_{01} : “I found the task requirements easy to understand”

Case	MM	MT	PI	ST
1	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
2	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
3	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
4	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
5	0 (0.0%)	0 (0.0%)	0 (0.0%)	1 (5.0%)
6	3 (15.0%)	3 (15.0%)	3 (15.0%)	2 (10.0%)
7	17 (85.0%)	17 (85.0%)	17 (85.0%)	17 (85.0%)
N	20	20	20	20
Median	7	7	7	7
Mean	6.85	6.85	6.85	6.8
S.D.	0.366	0.366	0.366	0.523

Tab. B.5: Q_{02} : “I found the task easy to achieve”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
2	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
3	0 (0.0%)	6 (30.0%)	0 (0.0%)	1 (5.0%)
4	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
5	1 (5.0%)	5 (25.0%)	0 (0.0%)	7 (35.0%)
6	4 (20.0%)	3 (15.0%)	5 (25.0%)	4 (20.0%)
7	15 (75.0%)	3 (15.0%)	15 (75.0%)	5 (25.0%)
N	20	20	20	20
Median	7	5	7	5
Mean	6.7	4.45	6.75	5.2
S.D.	0.571	1.761	0.444	1.673

Tab. B.6: Q_{03} : “I did most of the work in performing the task”

Case	MM	MT	PI	ST
1	4 (20.0%)	5 (25.0%)	4 (20.0%)	6 (30.0%)
2	1 (5.0%)	2 (10.0%)	1 (5.0%)	1 (5.0%)
3	4 (20.0%)	3 (15.0%)	2 (10.0%)	3 (15.0%)
4	9 (45.0%)	5 (25.0%)	10 (50.0%)	8 (40.0%)
5	1 (5.0%)	2 (10.0%)	2 (10.0%)	2 (10.0%)
6	1 (5.0%)	3 (15.0%)	1 (5.0%)	0 (0.0%)
7	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
N	20	20	20	20
Median	4	3.5	4	3.5
Mean	3.25	3.3	3.4	2.95
S.D.	1.410	1.780	1.465	1.468

Tab. B.7: Q_{04} : “I was distracted by what my partner was doing”

Case	MM	MT	PI	ST
1	4 (20.0%)	4 (20.0%)	5 (25.0%)	7 (35.0%)
2	4 (20.0%)	5 (25.0%)	5 (25.0%)	4 (20.0%)
3	3 (15.0%)	6 (30.0%)	5 (25.0%)	5 (25.0%)
4	2 (10.0%)	2 (10.0%)	2 (10.0%)	2 (10.0%)
5	3 (15.0%)	1 (5.0%)	2 (10.0%)	0 (0.0%)
6	2 (10.0%)	2 (10.0%)	1 (5.0%)	1 (5.0%)
7	2 (10.0%)	0 (0.0%)	0 (0.0%)	1 (5.0%)
N	20	20	20	20
Median	3	3	2.5	2
Mean	3.5	2.85	2.7	2.55
S.D.	2.039	1.531	1.490	1.701

Tab. B.8: Q_{05} : “My input was disrupted by what my partner was doing”

Case	MM	MT	PI	ST
1	7 (35.0%)	1 (5.0%)	7 (35.0%)	9 (45.0%)
2	6 (30.0%)	8 (40.0%)	7 (35.0%)	2 (10.0%)
3	2 (10.0%)	4 (20.0%)	4 (20.0%)	1 (5.0%)
4	0 (0.0%)	0 (0.0%)	1 (5.0%)	1 (5.0%)
5	3 (15.0%)	4 (20.0%)	0 (0.0%)	5 (25.0%)
6	1 (5.0%)	3 (15.0%)	1 (5.0%)	2 (10.0%)
7	1 (5.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
N	20	20	20	20
Median	2	3	2	2
Mean	2.65	3.35	2.15	2.85
S.D.	1.899	1.663	1.268	2.007

Tab. B.9: Q_{06} : “It was easy to use this interface with a partner”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	0 (0.0%)
2	0 (0.0%)	1 (5.0%)	1 (5.0%)	2 (10.0%)
3	2 (10.0%)	1 (5.0%)	0 (0.0%)	5 (25.0%)
4	2 (10.0%)	2 (10.0%)	1 (5.0%)	3 (15.0%)
5	3 (15.0%)	9 (45.0%)	3 (15.0%)	3 (15.0%)
6	5 (25.0%)	5 (25.0%)	8 (40.0%)	5 (25.0%)
7	8 (40.0%)	1 (5.0%)	7 (35.0%)	2 (10.0%)
N	20	20	20	20
Median	6	5	6	4.5
Mean	5.75	4.8	5.9	4.5
S.D.	1.372	1.436	1.252	1.606

Tab. B.10: Q_{07} : “It would be easier to use this interface by myself”

Case	MM	MT	PI	ST
1	2 (10.0%)	2 (10.0%)	2 (10.0%)	2 (10.0%)
2	4 (20.0%)	2 (10.0%)	5 (25.0%)	1 (5.0%)
3	2 (10.0%)	5 (25.0%)	3 (15.0%)	0 (0.0%)
4	5 (25.0%)	4 (20.0%)	5 (25.0%)	2 (10.0%)
5	3 (15.0%)	2 (10.0%)	1 (5.0%)	4 (20.0%)
6	1 (5.0%)	3 (15.0%)	2 (10.0%)	5 (25.0%)
7	3 (15.0%)	2 (10.0%)	2 (10.0%)	6 (30.0%)
N	20	20	20	20
Median	4	4	3.5	6
Mean	3.9	3.95	3.6	5.2
S.D.	1.917	1.820	1.847	1.936

Tab. B.11: Q_{08} : “I feel that I paid attention to what my partner was doing”

Case	MM	MT	PI	ST
1	3 (15.0%)	2 (10.0%)	1 (5.0%)	0 (0.0%)
2	3 (15.0%)	1 (5.0%)	7 (35.0%)	0 (0.0%)
3	4 (20.0%)	2 (10.0%)	1 (5.0%)	0 (0.0%)
4	0 (0.0%)	0 (0.0%)	1 (5.0%)	0 (0.0%)
5	5 (25.0%)	6 (30.0%)	2 (10.0%)	2 (10.0%)
6	4 (20.0%)	4 (20.0%)	6 (30.0%)	11 (55.0%)
7	1 (5.0%)	5 (25.0%)	2 (10.0%)	7 (35.0%)
N	20	20	20	20
Median	4	5	4.5	6
Mean	3.85	4.95	4.1	6.25
S.D.	1.954	1.959	2.075	0.639

Tab. B.12: Q_{09} : "I feel that my partner paid attention to me"

Case	MM	MT	PI	ST
1	3 (15.0%)	0 (0.0%)	1 (5.0%)	0 (0.0%)
2	4 (20.0%)	3 (15.0%)	4 (20.0%)	0 (0.0%)
3	2 (10.0%)	3 (15.0%)	1 (5.0%)	1 (5.0%)
4	3 (15.0%)	2 (10.0%)	4 (20.0%)	0 (0.0%)
5	4 (20.0%)	9 (45.0%)	4 (20.0%)	3 (15.0%)
6	2 (10.0%)	0 (0.0%)	4 (20.0%)	11 (55.0%)
7	2 (10.0%)	3 (15.0%)	2 (10.0%)	5 (25.0%)
N	20	20	20	20
Median	4	5	4.5	6
Mean	3.75	4.45	4.3	5.95
S.D.	1.970	1.572	1.809	0.945

Tab. B.13: Q_{10} : "I feel that I provided useful input to my partner"

Case	MM	MT	PI	ST
1	1 (5.0%)	2 (10.0%)	2 (10.5%)	1 (5.0%)
2	2 (10.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
3	1 (5.0%)	1 (5.0%)	2 (10.5%)	0 (0.0%)
4	5 (25.0%)	2 (10.0%)	1 (5.3%)	1 (5.0%)
5	9 (45.0%)	6 (30.0%)	5 (26.3%)	1 (5.0%)
6	2 (10.0%)	7 (35.0%)	5 (26.3%)	11 (55.0%)
7	0 (0.0%)	1 (5.0%)	4 (21.1%)	5 (25.0%)
N	20	20	19	20
Median	5	5	5	6
Mean	4.25	4.7	5	5.65
S.D.	1.333	1.720	1.856	1.599

Tab. B.14: Q_{11} : “I feel that my partner provided useful input”

Case	MM	MT	PI	ST
1	1 (5.0%)	0 (0.0%)	1 (5.0%)	1 (5.0%)
2	2 (10.0%)	2 (10.0%)	0 (0.0%)	0 (0.0%)
3	1 (5.0%)	1 (5.0%)	1 (5.0%)	0 (0.0%)
4	4 (20.0%)	4 (20.0%)	1 (5.0%)	2 (10.0%)
5	5 (25.0%)	6 (30.0%)	5 (25.0%)	2 (10.0%)
6	6 (30.0%)	6 (30.0%)	9 (45.0%)	10 (50.0%)
7	1 (5.0%)	1 (5.0%)	3 (15.0%)	5 (25.0%)
N	20	20	20	20
Median	5	5	6	6
Mean	4.6	4.8	5.4	5.7
S.D.	1.603	1.361	1.429	1.418

Tab. B.15: Q_{12} : “I feel that I could learn from observing my partner’s actions”

Case	MM	MT	PI	ST
1	4 (20.0%)	3 (15.0%)	4 (20.0%)	1 (5.0%)
2	0 (0.0%)	2 (10.0%)	3 (15.0%)	2 (10.0%)
3	7 (35.0%)	4 (20.0%)	1 (5.0%)	3 (15.0%)
4	3 (15.0%)	0 (0.0%)	4 (20.0%)	1 (5.0%)
5	3 (15.0%)	7 (35.0%)	4 (20.0%)	4 (20.0%)
6	3 (15.0%)	4 (20.0%)	4 (20.0%)	7 (35.0%)
7	0 (0.0%)	0 (0.0%)	0 (0.0%)	2 (10.0%)
N	20	20	20	20
Median	3	5	4	5
Mean	3.5	3.9	3.65	4.7
S.D.	1.670	1.804	1.872	1.780

Tab. B.16: Q_{13} : “I feel that I could demonstrate things to my partner”

Case	MM	MT	PI	ST
1	6 (30.0%)	2 (10.0%)	4 (20.0%)	3 (15.0%)
2	3 (15.0%)	4 (20.0%)	3 (15.0%)	2 (10.0%)
3	2 (10.0%)	4 (20.0%)	2 (10.0%)	3 (15.0%)
4	4 (20.0%)	3 (15.0%)	3 (15.0%)	3 (15.0%)
5	2 (10.0%)	3 (15.0%)	7 (35.0%)	2 (10.0%)
6	3 (15.0%)	2 (10.0%)	1 (5.0%)	6 (30.0%)
7	0 (0.0%)	2 (10.0%)	0 (0.0%)	1 (5.0%)
N	20	20	20	20
Median	3	3.5	4	4
Mean	3.1	3.75	3.45	4.05
S.D.	1.861	1.860	1.701	1.959

Tab. B.17: Q_{14} : “The application was stable”

Case	MM	MT	PI	ST
1	0 (0.0%)	0 (0.0%)	1 (5.0%)	0 (0.0%)
2	0 (0.0%)	0 (0.0%)	0 (0.0%)	1 (5.0%)
3	0 (0.0%)	4 (20.0%)	0 (0.0%)	1 (5.0%)
4	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
5	0 (0.0%)	5 (25.0%)	3 (15.0%)	2 (10.0%)
6	2 (10.0%)	6 (30.0%)	4 (20.0%)	7 (35.0%)
7	18 (90.0%)	4 (20.0%)	12 (60.0%)	8 (40.0%)
N	20	20	20	20
Median	7	5.5	7	6
Mean	6.9	5.25	6.2	5.85
S.D.	0.308	1.410	1.436	1.424

Tab. B.18: Q_{15} : “The application behaved predictably”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	3 (15.0%)
2	1 (5.0%)	1 (5.0%)	1 (5.0%)	0 (0.0%)
3	0 (0.0%)	9 (45.0%)	0 (0.0%)	3 (15.0%)
4	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
5	0 (0.0%)	4 (20.0%)	0 (0.0%)	4 (20.0%)
6	3 (15.0%)	4 (20.0%)	6 (30.0%)	8 (40.0%)
7	16 (80.0%)	0 (0.0%)	13 (65.0%)	1 (5.0%)
N	20	20	20	20
Median	7	3	7	5
Mean	6.6	3.9	6.45	4.55
S.D.	1.142	1.483	1.146	1.905

Tab. B.19: Q_{16} : “It was easy to move cards around the task space”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	2 (10.0%)
2	1 (5.0%)	4 (20.0%)	0 (0.0%)	1 (5.0%)
3	0 (0.0%)	8 (40.0%)	1 (5.0%)	5 (25.0%)
4	0 (0.0%)	0 (0.0%)	1 (5.0%)	4 (20.0%)
5	0 (0.0%)	5 (25.0%)	1 (5.0%)	4 (20.0%)
6	4 (20.0%)	2 (10.0%)	7 (35.0%)	4 (20.0%)
7	15 (75.0%)	0 (0.0%)	10 (50.0%)	0 (0.0%)
N	20	20	20	20
Median	7	3	6.5	4
Mean	6.55	3.5	6.2	3.95
S.D.	1.146	1.469	1.105	1.572

Tab. B.20: Q_{17} : “It was easy to correct any mistakes that were made”

Case	MM	MT	PI	ST
1	1 (5.0%)	1 (5.0%)	0 (0.0%)	2 (10.0%)
2	0 (0.0%)	4 (20.0%)	0 (0.0%)	1 (5.0%)
3	0 (0.0%)	3 (15.0%)	0 (0.0%)	5 (25.0%)
4	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
5	2 (10.0%)	5 (25.0%)	2 (10.0%)	5 (25.0%)
6	4 (20.0%)	5 (25.0%)	4 (20.0%)	6 (30.0%)
7	13 (65.0%)	1 (5.0%)	14 (70.0%)	0 (0.0%)
N	20	20	20	20
Median	7	5	7	5
Mean	6.3	4.2	6.6	4.2
S.D.	1.418	1.795	0.681	1.704

Tab. B.21: Q_{18} : “It was easy to recover from any errors in the interface”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
2	0 (0.0%)	3 (15.0%)	0 (0.0%)	2 (10.0%)
3	0 (0.0%)	2 (10.0%)	0 (0.0%)	2 (10.0%)
4	3 (15.0%)	4 (20.0%)	1 (5.0%)	3 (15.0%)
5	1 (5.0%)	6 (30.0%)	2 (10.0%)	6 (30.0%)
6	5 (25.0%)	3 (15.0%)	6 (30.0%)	5 (25.0%)
7	11 (55.0%)	1 (5.0%)	11 (55.0%)	1 (5.0%)
N	20	20	20	20
Median	7	4.5	7	5
Mean	6.2	4.2	6.35	4.5
S.D.	1.105	1.609	0.875	1.606

Tab. B.22: Q_{19} : “This method provides a good way to arrange information”

Case	MM	MT	PI	ST
1	0 (0.0%)	1 (5.0%)	0 (0.0%)	2 (10.0%)
2	1 (5.0%)	1 (5.0%)	0 (0.0%)	1 (5.0%)
3	0 (0.0%)	2 (10.0%)	1 (5.0%)	4 (20.0%)
4	0 (0.0%)	3 (15.0%)	5 (25.0%)	2 (10.0%)
5	6 (30.0%)	7 (35.0%)	2 (10.0%)	4 (20.0%)
6	2 (10.0%)	4 (20.0%)	6 (30.0%)	6 (30.0%)
7	11 (55.0%)	2 (10.0%)	6 (30.0%)	1 (5.0%)
N	20	20	20	20
Median	7	5	6	5
Mean	6.05	4.7	5.55	4.35
S.D.	1.317	1.559	1.317	1.785

Tab. B.23: Q_{20} : “This method felt like a natural way to solve the task”

Case	MM	MT	PI	ST
1	0 (0.0%)	3 (15.0%)	0 (0.0%)	4 (20.0%)
2	2 (10.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
3	2 (10.0%)	3 (15.0%)	1 (5.0%)	7 (35.0%)
4	0 (0.0%)	1 (5.0%)	3 (15.0%)	0 (0.0%)
5	4 (20.0%)	5 (25.0%)	2 (10.0%)	3 (15.0%)
6	4 (20.0%)	4 (20.0%)	4 (20.0%)	5 (25.0%)
7	8 (40.0%)	4 (20.0%)	10 (50.0%)	1 (5.0%)
N	20	20	20	20
Median	6	5	6.5	3
Mean	5.5	4.65	5.95	3.85
S.D.	1.732	2.033	1.317	1.981